

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## MONITOROVÁNÍ NÁVŠTĚVNÍKŮ WEBOVÝCH STRÁNEK

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

MARTIN JELIČ

BRNO 2011



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# **MONITOROVÁNÍ NÁVŠTĚVNÍKŮ WEBOVÝCH STRÁNEK**

WEB PAGE VISITOR MONITORING

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MARTIN JELIČ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. RADEK BURGET, Ph.D.**

BRNO 2011

## Abstrakt

Práce se zabývá webovou analytikou, jejími pojmy, principy, souvisejícími problémy a jejich řešením. Je zde zevrubně popsáno několik existujících nástrojů pro webovou analytiku. Těžištěm práce je návrh a implementace nového nástroje, který umožňuje monitorovat provoz na webových stránkách a vyhodnocovat údaje za účelem řízení internetových projektů. V rámci práce jsou prezentovány výsledky testování nástroje v reálném provozu a jejich srovnání s existujícími nástroji, od kterých se nový nástroj v některých vlastnostech odlišuje. Práce pojednává též o výhodách použití dokumentově orientované databáze MongoDB pro účely monitorování návštěvnosti webových stránek.

## Abstract

This bachelor's thesis deals with web analytics, its terms, principles, related problems and their solutions. There are described at large a few tools for web analytics. The focus of this work is the design and implementation of a new web analytics tool that allows to monitor the web page traffic and evaluate the obtained data for internet project management purposes. There are shown the results of testing the mentioned tool in real traffic and the comparison with the results of the existing tools, which differ from the new tool in some characteristics. The work discusses the advantages of using the MongoDB document-oriented database for website traffic monitoring purposes as well.

## Klíčová slova

webová analytika, měření návštěvnosti webových stránek, konverze, konverzní poměr, politika stejného původu, MongoDB, MapReduce

## Keywords

web analytics, website traffic measuring, conversion, conversion rate, same-origin policy, MongoDB, MapReduce

## Citace

Martin Jelič: Monitorování návštěvníků  
webových stránek, bakalářská práce, Brno, FIT VUT v Brně, 2011

# Monitorování návštěvníků webových stránek

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Radka Burgeta, Ph.D. a uvedl všechnu použitou literaturu a prameny.

.....  
Martin Jelič  
17. května 2011

## Poděkování

Děkuji vedoucímu bakalářské práce panu Ing. Radku Burgetovi, Ph.D. za odborné vedení, cenné připomínky a osobní konzultace během tvorby této práce.

© Martin Jelič, 2011.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Cíl práce . . . . .	3
1.2	Motivace . . . . .	4
1.3	Přehled kapitol . . . . .	4
<b>2</b>	<b>Webová analytika</b>	<b>5</b>
2.1	Definice pojmů . . . . .	5
2.1.1	Načtení prvku / Hit (Hit) . . . . .	6
2.1.2	Zobrazení stránky (Page view) . . . . .	6
2.1.3	Návštěva (Visit) . . . . .	6
2.1.4	První návštěva (First visit) . . . . .	6
2.1.5	Návštěvník (Visitor) . . . . .	6
2.1.6	Náhled / odraz (Singleton / Bounce) . . . . .	7
2.1.7	Míra opuštění / míra odrazu (Bounce rate) . . . . .	7
2.1.8	Odkazující stránka (Referrer) . . . . .	7
2.1.9	Zdroj návštěvy (Source of visit) . . . . .	7
2.1.10	Konverze (Conversion) . . . . .	7
2.1.11	Konverzní poměr (Conversion rate) . . . . .	7
2.1.12	Cílová stránka (Landing page) . . . . .	8
2.2	Způsoby měření . . . . .	8
2.2.1	Analýza logů na serveru . . . . .	8
2.2.2	Měřicí kód ve stránce . . . . .	9
2.3	Úskalí a problémy měření . . . . .	10
2.3.1	Prostředky pro identifikaci uživatele . . . . .	10
2.3.2	Podpora JavaScriptu v prohlížečích . . . . .	11
2.3.3	Bezpečnostní politika prohlížečů — otázka stejného původu (same-origin policy) . . . . .	11
2.3.4	Přístupy robotů . . . . .	12
<b>3</b>	<b>Nástroje pro analýzu návštěvnosti</b>	<b>13</b>
3.1	Google Analytics . . . . .	13
3.2	Yahoo Web Analytics . . . . .	13
3.3	Adobe SiteCatalyst . . . . .	14
3.4	AWStats . . . . .	15
3.5	Webalizer . . . . .	15
3.6	TopList.cz . . . . .	16
3.7	mYx . . . . .	16
3.8	CrazyEgg . . . . .	16

3.9	Shrnutí . . . . .	17
<b>4</b>	<b>Knihovna/nástroj pro sledování zdrojů a pohybu návštěvníků webu</b>	<b>18</b>
4.1	Specifikace požadavků . . . . .	18
4.2	Název knihovny/nástroje . . . . .	18
4.3	Použité technologie . . . . .	18
4.3.1	Nette framework . . . . .	18
4.3.2	MongoDB . . . . .	19
4.4	Návrh . . . . .	20
4.4.1	Sběr dat . . . . .	22
4.4.2	Ukládání dat . . . . .	23
4.4.3	Analýza dat . . . . .	23
4.4.4	Prezentace výsledků . . . . .	23
4.5	Implementace . . . . .	25
4.5.1	Sběr a ukládání dat . . . . .	25
4.5.2	Analýza dat . . . . .	26
4.5.3	Prezentace výsledků . . . . .	28
<b>5</b>	<b>Nástroj pro měření účinnosti internetové reklamy</b>	<b>30</b>
5.1	Metodika měření . . . . .	30
5.2	Implementace . . . . .	32
5.2.1	Měření konverzí . . . . .	32
5.2.2	Rozpoznávání návštěvníků z reklamní kampaně . . . . .	32
<b>6</b>	<b>Testování v reálném prostředí</b>	<b>33</b>
6.1	Web s vyšší návštěvností . . . . .	33
6.2	Web s nižší návštěvností . . . . .	34
<b>7</b>	<b>Závěr</b>	<b>38</b>
<b>A</b>	<b>Obsah CD</b>	<b>41</b>
<b>B</b>	<b>Požadavky pro chod knihovny/nástroje MJmonitor</b>	<b>42</b>

# Kapitola 1

## Úvod

Se stále rostoucím využitím internetu pro komerční účely (reklama, obchod, služby, poskytování obsahu atd.) roste i potřeba co nejpodrobněji monitorovat návštěvníky webových stránek. Pro efektivní podnikání, ať už na internetu, či mimo něj, je nutné sledovat zpětnou vazbu. V kamenném obchodě nás například zajímá kolik lidí přijde, co si tito lidé prohlíží, jaké procento návštěvníků si něco zakoupí apod. U reklamy na billboardu nás pravděpodobně bude zajímat, kolik automobilů projede kolem. U televizních programů měříme za pomoci tzv. peplemeterů počet diváků a jejich demografické složení.

Internet poskytuje rozsáhlé spektrum možností pro získávání zpětné vazby, a těmito možnostmi se zabývá obor, nazývaný webová analytika. Lze automatizovaně měřit počet návštěvníků, sledovat jejich pohyb v rámci webu, analyzovat čas, který stráví na jednotlivých stránkách. Lze rovněž zaznamenávat, odkud návštěvníci na web přicházejí a na kterých stránkách web opouštějí. Je také možné počítat, kolik procent návštěvníků splnilo určitý cíl (provedení nějaké akce) a jaká cesta k tomuto cíli nejčastěji vedla. Množství prvků, které lze monitorovat, je velké. Ne všechny jsou však v praxi smysluplně využitelné a je též potřeba zmínit, že monitorování návštěvníků webových stránek skýtá i mnoho úskalí, se kterými se musíme vypořádat. Sestavení měřicích algoritmů a definování výsledků měření je často věcí kompromisů.

Efektivním využitím možností, které nám internet pro monitoring návštěvníků nabízí, i řešení různých souvisejících problémů se zabývá právě tato práce.

### 1.1 Cíl práce

Cílem této bakalářské práce je zmapovat situaci v oblasti webové analytiky, shromáždit informace o dostupných prostředcích pro monitorování návštěvníků webových stránek a vytvořit nástroj, který umožní:

1. Tvůrcům webových stránek integrovat do jimi tvořených webů knihovnu, umožňující monitorování návštěvníků a vyhodnocování získaných údajů.
2. Provozovatelům webových stránek vyhodnocovat úspěšnost jejich webů a efektivitu použitých internetových reklamních prostředků.

## 1.2 Motivace

Webová analytika se dá v současné době označit za stabilizovaný obor, který má jasné stanovené hranice, dané především technickými možnostmi měření. Trh nástrojů pro webovou analytiku je rovněž poměrně stabilní — dominantním nástrojem pro laiky i odborníky je služba Google Analytics, která je úspěšná zejména pro svoje rozsáhlé spektrum možností, dostupnost, cenovou politiku (použití je volné a zdarma) a v neposlední řadě díky značce Google. Pro odborníky a rozsáhlé projekty existuje několik dalších nástrojů, jako např. Adobe/Omniture SiteCatalyst, Coremetrics, nebo Yahoo Web Analytics. Pro laiky jsou k dispozici další, velmi jednoduché nástroje — v ČR je to např. TopList.cz, nebo Navrcholu.cz. Podrobněji o nástrojích pro webovou analytiku pojednává kapitola 3.

Na základě svých zkušeností, získaných v oboru internetového marketingu, se domnívám, že na současném trhu nástrojů pro webovou analytiku chybí nástroj jakési střední kategorie, který by dokázal provozovateli webu poskytnout dostatek důležitých podkladů pro rozhodování o vývoji webu a formách jeho propagace, avšak v jednoduché a srozumitelné formě, přímo, bez nutnosti složitého nastavování pohledů na data, sestav, segmentů, apod. V současnosti nejpoužívanější nástroj Google Analytics, který je používán odborníky i laiky, umí téměř vše potřebné, avšak zorientovat se v jeho spektru možností a nastavení není triviální záležitost. Skutečnou „sílu“ takového nástroje dokáží efektivně využít zpravidla pouze odborníci v oboru webové analytiky. Na trhu existují samozřejmě i jednoduché a snadno srozumitelné nástroje, jmenujme například český TopList. Tento typ nástrojů však většinou neposkytuje dostatečně komplexní podklady pro rozhodování — chybí zde měření konverzí, měření zdrojů návštěvnosti bývá značně omezené, nelze podrobněji sledovat chování konkrétních návštěvníků.

Tato bakalářská práce se tedy ve své praktické části snaží o návrh nástroje, který by dokázal poskytnout dostatek údajů pro řízení internetového byznysu, ale který by byl zároveň jednoduchý, oproštěný od méně důležitých údajů, soustředil se na to nejpodstatnější a co nejvíce by provozovateli webu usnadnil cestu k údajům. V ideálním případě by tento nástroj měl umožnit provozovateli webu činit rozhodnutí a sledovat jejich následky bez nutnosti poradenství externí agentury.

## 1.3 Přehled kapitol

V kapitole 2 probereme obor webové analytiky jako celek, jeho strukturu, základní pojmy a také konkrétní používané techniky, vč. jejich výhod a nevýhod. Součástí kapitoly je i přehled nejznámějších problémů, s nimiž se v oboru můžeme setkat.

Kapitola 3 se snaží zevrubně zmapovat současný trh nástrojů, určených pro webovou analytiku. Protože nástrojů existuje velké množství, text se zabývá jen malou částí. Popisuje světově nejznámější komplexní nástroje jako jsou Google Analytics nebo Adobe/Omniture SiteCatalyst, ale nechybí ani popis malého českého projektu TopList, nebo úzce zaměřených nástrojů mYx a CrazyEgg.

Další dvě kapitoly, označené čísly 4 a 5, se již zabývají návrhem a implementací vlastního nástroje pro webovou analytiku.

Číslo 6 patří kapitole, představující testování vzniklého nástroje v reálném provozu. Nechybí zde srovnání výsledků s ostatními nástroji.

Závěrečná kapitola provádí zhodnocení této práce jako celku, dosažených výsledků a přínosů a nastiňuje možný budoucí vývoj vzniklého nástroje.



## Kapitola 2

# Webová analytika

Definice mezinárodní organizace Web Analytics Association [8] říká, že webová analytika je měření, sběr, analýza a reportování internetových dat za účelem porozumění a optimalizace užívání webu.

Zjednodušeně by se dalo říci, že webová analytika poskytuje tvůrcům, provozovatelům a propagátorům webových stránek zpětnou vazbu od uživatelů. Základem webové analytiky je měření návštěvnosti webových stránek a monitorování chování jejích uživatelů.

Webová analytika se dá rozdělit do tří fází:

1. Sběr a shromažďování dat
2. Analýza dat
3. Presentace (reportování) dat

V praxi se pro webovou analytiku používají speciální nástroje, které se většinou starají o všechny tři fáze (existujícím nástrojům se věnuje kapitola 3). V rámci této kapitoly si nejprve ujasníme základní pojmy, které s webovou analytikou souvisí, dále probereme základní způsoby měření, používané nástroje pro webovou analytiku a na závěr kapitoly se budeme zabývat problémy a úskalími, na která webová analytika v praxi naráží.

### 2.1 Definice pojmů

S webovou analytikou je spojeno mnoho pojmů, jejichž význam není z termínu samotného zřejmý a rovněž definice některých pojmů je sporná. Považuji za vhodné některé pojmy vysvětlit a definovat jejich význam v rámci této práce. V oboru webové analytiky zatím neexistují standardy nebo normy, které by se zabývaly definováním pojmů. Určitými autoritami v této oblasti jsou organizace JICWEBS (The Joint Industry Committee for Web Standards), ABCe (Audit Bureau of Circulations electronic), WAA (Web Analytics Association) a IAB (Interactive Advertising Bureau). Na základě doporučení těchto organizací byla zpracována stránka na anglické verzi Wikipedie [10], ze které definice některých pojmů v této práci vycházejí.

Za českým výrazem je vždy uvedena anglická verze, neboť české překlady jsou v některých případech velmi volné.

### 2.1.1 Načtení prvku / Hit (Hit)

Jedná se o načtení jednoho samostatného souboru ze serveru. Typicky se jedná o prvky webové stránky — obrázky, soubory s definicí kaskádových stylů, soubory se zdrojovými kódy JavaScriptu, apod. Tento údaj u některých typů nástrojů pro webovou analytiku není dostupný (více viz kapitola 2.2 na straně 8). V takovém případě se však nejedná o zásadní nevýhodu, neboť tento údaj nevypovídá téměř nic o chování návštěvníků na webu a pro primární účely webové analytiky není příliš významný. Může být užitečný pro optimalizaci rychlosti načítání webových stránek, neboť velký počet načítaných prvků v rámci jednoho zobrazení stránky (viz níže) může zpomalit její načtení.

### 2.1.2 Zobrazení stránky (Page view)

Jedná se o načtení jedné samostatné webové stránky. V praxi tedy jde o načtení souboru ze serveru, přičemž soubor musí být typu, který je definován jako stránka. Součástí zobrazení stránky je načtení jednotlivých prvků stránky, jedno zobrazení stránky tedy obvykle vygeneruje několik hitů (viz výše).

### 2.1.3 Návštěva (Visit)

Návštěva je definována jako posloupnost požadavků (zobrazení stránky — viz výše) od stejného, unikátně identifikovaného uživatele, přičemž časová prodleva mezi jednotlivými požadavky nepřesahuje 30 minut.

Pokud tedy přijde na web nový uživatel, jedná se o novou návštěvu. Pokud si následně tento uživatel prohlíží stránky webu a mezi jednotlivými zobrazeními stránek neuplyne více než 30 minut, jedná se stále o 1 návštěvu. Pokud poté uživatel odejde od počítače, vrátí se za více než 30 minut a provede zobrazení stránky z téhož webu, započítá se nová (druhá) návštěva uživatele.

### 2.1.4 První návštěva (First visit)

Návštěva od uživatele, který web nikdy dříve nenavštívil. Význam slova „nikdy“ je omezen měřeným časovým úsekem a rovněž platností prostředků pro rozpoznávání unikátních návštěvníků (více v kapitole 2.2 na straně 8).

### 2.1.5 Návštěvník (Visitor)

Obecně rozšířená definice říká, že návštěvník je unikátně identifikovaný uživatel webu. Ve skutečnosti se jedná spíše o unikátně identifikovaný počítač, potažmo internetový prohlížeč. V rámci anonymního monitorování návštěvnosti webových stránek je velice obtížné (dá se říci, že nemožné) rozlišovat návštěvníky skutečně ve smyslu osob. Pokud člověk otevře ten samý web na dvou počítačích, systém jej zaznamená jako dva návštěvníky. Ani unikátnost počítače nebo prohlížeče nelze stoprocentně zajistit, neboť identifikace naráží na technická omezení (blíže o tomto problému pojednává kapitola 2.2 na straně 8)

V průběhu měření běžně jeden návštěvník provede několik návštěv a v rámci nich několik zobrazení stránek.

### 2.1.6 Náhled / odraz (Singleton / Bounce)

Návštěva, skládající se pouze z jednoho zobrazení stránky. Většinou indikuje, že návštěvníka na stránce nic nezaujalo a proto web ihned opustil.

### 2.1.7 Míra opuštění / míra odrazu (Bounce rate)

Tento údaj označuje procento návštěv, při kterých návštěvník opustí web hned na první stránce, kterou navštívil (tzn. nenavštíví žádné jiné stránky webu).

### 2.1.8 Odkazující stránka (Referrer)

Označuje se takto stránka, ze které uživatel přišel na sledovanou stránku. Např. uživatel si prohlíží stránku *www.priklad.cz/prvni-stranka* a klikne na odkaz „Druhá stránka“, který ho přivede na stránku *www.priklad.cz/druha-stranka*. Pokud sledujeme druhou stránku, odkazující stránkou je v tomto případě první stránka.

### 2.1.9 Zdroj návštěvy (Source of visit)

Jedná se o způsob, jakým se uživatel dostal na sledovaný web. Základní způsoby jsou pouze dva — přímá návštěva a návštěva z odkazující stránky. Přímou návštěvou se rozumí přímé zadání URL adresy do adresního řádku prohlížeče. V praxi se jako přímá návštěva započítává i kliknutí na záložku (oblíbenou položku) v prohlížeči, kliknutí na tlačítko „domů“ a jakákoliv jiná návštěva, u které nelze rozpoznat odkazující zdroj (např. neoznačený odkaz v lokálně uloženém PDF dokumentu).

Návštěvy z odkazujících stránek se obvykle rozdělují na návštěvy z vyhledávačů a návštěvy z ostatních webů. Vyhledávače mají oproti ostatním odkazujícím stránkám jedno specifikum — obvykle se u nich sledují klíčová slova, pomocí kterých uživatel sledovaný web našel. V rámci této práce jsou odděleny pouze nejvýznamnější české (Seznam, Centrum, Atlas a Jyxo) a světové (Google, Bing a Yahoo) vyhledávače.

V rámci výsledků měření lze zdroje návštěvnosti dále dělit pomocí značek v URL. Díky této metodě se dá rozpoznat např. návštěva z placené reklamní kampaně.

### 2.1.10 Konverze (Conversion)

Konverze je událost, při které návštěvník (viz výše) provede nějakou předem definovanou akci, jinými slovy splní konverzní cíl. Typicky se konverze sledují u internetových obchodů, kde je konverzním cílem provedení objednávky. Dalším příkladem je provedení on-line rezervace na webu hotelu. V takových případech můžeme konverzi definovat jako událost, při které se návštěvník mění v zákazníka. Vždy to ale neplatí — jako konverzi můžeme označit např. odeslání kontaktního formuláře, registraci, přihlášení k odběru newsletteru, kliknutí na tlačítko „Líbí se mi“ u plug-inů sociální sítě Facebook, kliknutí na odkaz s affiliate programem, apod.

V mnoha případech se dá konverzi přiřadit určitá hodnota (např. zisk z provedené objednávky).

### 2.1.11 Konverzní poměr (Conversion rate)

Konverzní poměr vyjadřuje pravděpodobnost, že návštěvník webu provede konverzi. Vychází z naměřených údajů a počítá se jako poměr počtu provedených konverzí vůči počtu

návštěv, případně jako poměr počtu návštěvníků, kteří provedli alespoň jednu konverzi vůči celkovému počtu návštěvníků.

### 2.1.12 Cílová stránka (Landing page)

Jedná se o stránku, která je cílem určité reklamní kampaně — dostanou se na ní návštěvníci např. kliknutím na reklamní banner, inzerát v PPC systému, odkaz v reklamním e-mailu apod. Cílová stránka má za úkol co nejlépe nasměrovat návštěvníka ke konverzi.

## 2.2 Způsoby měření

K získávání dat o návštěvnosti a návštěvnících webových stránek se používají dva základní přístupy:

**site-centric** Získávání dat na straně měřené stránky. Nevyžaduje ze strany uživatele žádnou interakci, uživatel o něm většinou neví.

**user-centric** Získávání dat na straně uživatele, nebo jeho internetového prohlížeče. Vyžaduje určitou interakci uživatele, např. instalaci doplňku prohlížeče, nebo vyplnění dotazníku.

Zmíněné dva přístupy lze i kombinovat<sup>1</sup>. V rámci této práce se budeme zabývat pouze prvním přístupem. Ten můžeme dále rozdělit z hlediska technologického přístupu k problému měření údajů na dvě základní kategorie, jimiž se zabývají následující dvě podkapitoly.

### 2.2.1 Analýza logů na serveru

První přístup využívá logovacích souborů serveru. Tyto soubory zpravidla uchovávají informace o historii provedených požadavků na server zvenčí. U webového serveru se typicky jedná o požadavky na zobrazení webové stránky. Informace o každém požadavku může obsahovat několik údajů — např. IP adresu klienta, adresu požadované stránky, adresu odkazující stránky, HTTP kód, datum a čas operace nebo informace o prohlížeči klienta (příklad záznamu v logovacím souboru uvádí ukázka kódu 2.1). Pomocí různých agregačních operací nad údaji tohoto typu pro větší množství požadavků je možné sestavit souhrnné statistiky za určitá období — např. počet zobrazení stránek, počet návštěv, počet unikátních návštěvníků, nejčastější zdroje návštěv, nejčastěji používané typy prohlížečů, atd.

Protože sběr dat v tomto případě probíhá automaticky (stará se o něj webový server), nástroje, využívající tento způsob nemusí řešit fázi sběru a shromažďování dat a zabývají se pouze jejich analýzou a prezentací.

```
90.177.151.62 - - [29/Aug/2009:14:10:13 +0200] "GET / HTTP/1.1" 200 1271 "-"  
"Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.1.2) Gecko/20090729  
Firefox/3.5.2"
```

*Kód 2.1: Ukázka záznamu v logu webového serveru*

<sup>1</sup>V České republice používá kombinovaný přístup k získávání dat služba NetMonitor, kterou provozuje Sdružení pro internetovou reklamu [3].

### 2.2.2 Měřicí kód ve stránce

Druhý, v současné době více používaný způsob měření vyžaduje vložení tzv. měřicího kódu do všech stránek webu, které chceme měřit. Měřicí kód se stará o získání potřebných údajů (o návštěvníkovi a jeho přístupu) a zároveň zajišťuje předání informací monitorovacímu systému. Tento způsob měření má oproti analýze logů na serveru několik výhod:

- Nezávislost na typu webového serveru. Není potřeba řešit různé formáty serverových logů.
- Je možné provozovat měřicí nástroj na jiném serveru, než na kterém je umístěn měřený web. Díky tomu je možné sledovat současně několik webů, rozmístěných po různých serverech.
- Automatická ignorace většiny internetových robotů. Roboti totiž zpravidla nedisponují interpretem jazyka JavaScript, na němž bývá měřicí kód nejčastěji postaven.

Měřicí kód může mít různé podoby. Některé jednodušší systémy používají HTML značku `<img>`, která volá serverový skript měřicího systému. Ten zaznamená potřebné údaje a vygeneruje obrázek (buď viditelný ve formě počítadla, nebo „neviditelnou“ jednopixelovou tečku). Tento způsob používá například česká služba TopList. Výhodou je, že jsou zaznamenáni i návštěvníci s vypnutou podporou jazyka JavaScript v prohlížeči. Nicméně bez JavaScriptu není možné získat podrobnější údaje o návštěvníkovi (např. z jakého zdroje na stránku přišel), takže většina systémů tento přístup stejně kombinuje s JavaScriptovým kódem.

Běžnějším přístupem je použití samostatného kódu v jazyce JavaScript, který se postará o identifikaci návštěvníka, získání potřebných údajů a jejich předání serveru. Z atributů různých objektů lze v JavaScriptu zjistit mnoho informací o návštěvníkovi (z jakého zdroje na stránku přišel, jaký má prohlížeč, operační systém, rozlišení obrazovky, apod.) a zobrazené stránky (URL, titulek, apod.). Použití JavaScriptového měřicího kódu má i svá úskalí, která jsou rozebrána v následující kapitole.

```
<script type="text/javascript">
  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', 'UA-1234567-1']);
  _gaq.push(['_trackPageview']);

  (function() {
    var ga = document.createElement('script');
    ga.type = 'text/javascript';
    ga.async = true;
    ga.src = ('https:' == document.location.protocol ?
      'https://ssl' : 'http://www')
      + '.google-analytics.com/ga.js';
    var s~ = document.getElementsByTagName('script')[0];
    s.parentNode.insertBefore(ga, s);
  })();
</script>
```

Kód 2.2: Ukázka měřicího kódu v jazyce JavaScript — Google Analytics

```

```

Kód 2.3: Ukázka měřicího kódu ve formě obrázku — TopList

```

<a href="http://www.toplist.cz/">
  <script language="JavaScript" type="text/javascript">
    <!--
      document.write('');
    //-->
  </script>
  <noscript>
    
  </noscript>
</a>

```

Kód 2.4: Ukázka měřicího kódu ve formě kombinace obrázku s JavaScriptem — TopList

## 2.3 Úskalí a problémy měření

Obecně největším problémem měření návštěvnosti webových stránek je identifikace návštěvníků. Jak již bylo zmíněno výše, v rámci anonymního měření je v podstatě nemožné identifikovat unikátní skutečnou osobu. Většinou dokážeme identifikovat jednotlivé prohlížeče v rámci počítačů, v lepším případě jednotlivý počítač (přesněji řečeno operační systém). Přesto je zvykem naměřeným číslům přisuzovat jednotky typu „počet návštěvníků“, nebo „počet unikátních návštěvníků“.

### 2.3.1 Prostředky pro identifikaci uživatele

V počátcích rozvoje internetu měla v rámci identifikace uživatelů (počítačů) velkou vypovídací hodnotu IP adresa. Většina počítačů měla unikátní IP adresu a protože tuto adresu není problém zjistit, bylo možné ji použít jako rozlišovací klíč. V současné době je IP adresa k identifikaci uživatele téměř nepoužitelná, neboť existuje mnoho podsítí (s více počítači), které díky technologii NAT<sup>2</sup> v rámci internetu vystupují pod jednou IP adresou.

Identifikace uživatele se v současné době řeší převážně pomocí cookies<sup>3</sup>, které bývají někdy doplněny dalšími údaji. Při první návštěvě uživatele vygenerujeme unikátní kód, který uložíme do cookie. Při dalších návštěvách prohlížeč uživatele tento kód odešle serveru, který podle něj uživatele rozpozná.

Problém je v tom, že prohlížeče disponují možností vymazat cookies, případně zcela vypnout nebo omezit jejich podporu. Tento fakt zkresluje měření, neboť uživatele, kteří odstraňují cookies a uživatele s vypnutou podporou cookies může systém zaznamenat vícenásobně. Význam tohoto problému však není zásadní, neboť podíl uživatelů, kteří mají vypnutou podporu cookies, je velmi malý. Konkrétně v České republice vydal v roce 2007 server NAVRCHOLU.cz tiskovou zprávu [9], ve které uvádí, že v rámci webů, monitorovaných stejnojmennou službou, dosahuje podíl návštěv uživatelů s vypnutými cookies 0,46 %, přičemž dlouhodobý trend prokazuje mírný kontinuální pokles tohoto čísla.

<sup>2</sup>NAT — Network Address Translation — způsob mapování IP adres z jedné skupiny na jinou. Používá se většinou pro přístup více počítačů z lokální sítě na Internet pod jednou veřejnou adresou.

<sup>3</sup>HTTP cookie — data, která webový server pošle prohlížeči, ten je uloží na počítači uživatele a při každé další návštěvě téhož serveru prohlížeč tato data posílá serveru zpět. Slouží většinou k ukládání uživatelského nastavení a k identifikaci uživatelů.

Navíc i uživatele s vypnutými cookies lze do určité míry identifikovat. Webové prohlížeče totiž umožňují protistraně zjistit mnoho informací o prohlížeči samotném, o jeho verzi, nastavení, nainstalovaných zásuvných modulech, ale částečně i o nastavení operačního systému počítače (zde se jedná např. o rozlišení obrazovky či barevnou hloubku). Vzhledem k tomu, že informací je mnoho, dá se předpokládat, že existuje velmi málo počítačů, které se budou ve 100 % informací shodovat. Proto lze tato data v souhrnu použít pro identifikaci návštěvníka. Vzhledem k jisté analogii s lidskými otisky prstů jsou zmíněná data v angličtině označována jako „device fingerprints“. Narozdíl od lidských otisků prstů mají tyto „strojové otisky“ nevýhodu, že se mohou často měnit. Americká organizace Electronic Frontier Foundation provozuje experimentální projekt s názvem Panopticlick, jehož úkolem je zjistit použitelnost „strojových otisků“ pro identifikaci uživatelů. Ze statistik jejich výzkumu [5] vychází, že mezi prohlížeči s podporou platformy Flash nebo Java je cca 94 % otisků unikátních. Experiment se zároveň zabýval frekvencí změny otisků, přičemž zjistil, že otisky se sice mění často, ale za pomoci jednoduché heuristiky lze zjistit, který otisk je nový a který se pouze změnil. Úspěšnost této heuristiky v rámci experimentu se pohybovala kolem 99 %.

### 2.3.2 Podpora JavaScriptu v prohlížečích

Pokud je pro zaznamenávání přístupů návštěvníků využita technologie měřicího kódu v jazyce JavaScript, znamená to, že pokud prohlížeč uživatele nedisponuje podporou tohoto jazyka (případně má podporu vypnutou), přístup uživatele na web se nezaznamená. Ve výše zmíněné zprávě serveru NAVRCHOLU.cz [9] je uveden 1,19 % podíl uživatelů s vypnutým JavaScriptem. Je tedy potřeba brát v úvahu, že tato část uživatelů není započítána do statistik. Vzhledem k stále se rozšiřující technologii AJAX lze však předpokládat, že tento podíl se neustále zmenšuje.

### 2.3.3 Bezpečnostní politika prohlížečů — otázka stejného původu (same-origin policy)

Tento problém se týká rovněž měřicích systémů, které využívají technologii měřicího kódu v jazyce JavaScript. Pro komunikaci mezi webovou stránkou v klientském prohlížeči a serverovými skripty se používá obvykle objekt *XmlHttpRequest*, který je pro tyto účely určen. V případě, že potřebujeme komunikovat se serverovými skripty, které se nacházejí na jiném serveru, než webová stránka, ze které komunikace pochází, narazíme na bezpečnostní omezení prohlížečů, tzv. same-origin policy. Prohlížeče v rámci objektu *XmlHttpRequest* povolují pouze komunikaci se stejnou doménou, odkud pochází webová stránka, ze které metody objektu voláme. Pokud se snažíme komunikovat se skriptem na jiné doméně, prohlížeč spojení zablokuje.

Toto omezení se dá však jednoduše obejít. Známý je způsob, který bývá označován jako JSON-P<sup>4</sup>. Místo objektu *XmlHttpRequest* zde pracujeme s HTML/XHTML elementem `<script>`. Princip je následující: Pomocí metod pro manipulaci s DOM<sup>5</sup>, nebo pomocí metody `document.write()`, vložíme do zdrojového kódu stránky element `<script>`, jehož atribut `src` obsahuje URL adresu serverového skriptu. Data, která potřebujeme poslat na server, přidáme na konec URL ve formě query stringu. Pokud potřebujeme v klientském

<sup>4</sup>JSON with Padding — toto označení není příliš výstižné, neboť ve skutečnosti nemusí mít tato metoda vůbec nic společného s JSON.

<sup>5</sup>DOM — Document Object Model — objektově orientovaná reprezentace XML nebo HTML dokumentu. Umožňuje dynamickou manipulaci s obsahem dokumentu.

skriptu získat a zpracovat následnou odpověď serveru, vytvoříme callback funkci, jejímž argumentem bude právě požadovaná odpověď serveru. Uvnitř funkce potom s touto odpovědí (argumentem funkce) provedeme, co potřebujeme. Název callback funkce serveru předáme rovněž formou query stringu. Server tedy při zavolání dostane na vstup data metodou GET, s těmito daty provede potřebné operace a pokud má k dispozici název callback funkce, pošle klientovi odpověď ve formě *nazevFunkce(odpověď)*. Ve zdrojovém kódu webové stránky v klientském prohlížeči tedy potom vznikne podobný blok, jako v ukázce 2.5.

```
<script type="text/javascript"
  src="http://www.domena.cz/zpracuj.php?param1=data&param2=data_
  &callback=zpracujOdpoved">
  zpracujOdpoved('odpoved serveru');
</script>
```

Kód 2.5: Příklad výsledného HTML/JavaScript kódu, který vzniká při použití metody JSON-P

Server svou odpověď „obalí“ do volání specifikované callback funkce v jazyce JavaScript. Funkce může odpověď libovolným způsobem zpracovat, například ji uložit do cookies, nebo zobrazit návštěvníkovi.

### 2.3.4 Přístupy robotů

Je potřeba si uvědomit, že web nenavštěvují jen lidé prostřednictvím svých internetových prohlížečů, ale i různé automaticky fungující počítačové programy. Obvykle se označují jako roboti a slouží zpravidla ke sběru různých dat. Nejznámější jsou roboti internetových vyhledávačů (Googlebot, SeznamBot, apod.), které pomocí hypertextových odkazů prohledávají internet a zaznamenávají si potřebné informace o webových stránkách do své databáze.

U měřicích systémů, které fungují na principu analýzy log souboru webového serveru, je nutno počítat s tím, že v log souboru jsou zaznamenány všechny přístupy klientů na server, včetně přístupů robotů. V rámci webové analytiky nás ale zajímají samostatné výsledky pro provoz generovaný lidmi (případně i samostatné výsledky pro provoz generovaný roboty, ale ne obojí dohromady). Je tedy potřeba přístupy robotů odfiltrovat. K tomu slouží různé databáze robotů.

Měřicí systémy fungující na principu měřicího kódu v jazyce JavaScript mají situaci o něco jednodušší. Většina robotů totiž neumí kód v jazyce JavaScript vykonat, takže jejich přístup se do statistik vůbec nezaznamená. Existují ale i výjimky, např. roboty vyhledávačů, sloužící ke generování screenshotů stránek. Takové roboty musí umět spouštět všechny prvky stránky, jako běžné prohlížeče, proto vykonávají i kód v JavaScriptu a měřicí systém jejich přístupy zaznamená. Opět je tedy potřeba je nějakým způsobem odfiltrovat.



## Kapitola 3

# Nástroje pro analýzu návštěvnosti

### 3.1 Google Analytics

Google Analytics je v současné době nejrozšířenější nástroj pro analýzu návštěvnosti webových stránek [11] [12]. Jedná se o hostovanou webovou službu (SaaS<sup>1</sup>), jejíž použití je zdarma. Je určen širokému spektru uživatelů, laikům i odborníkům.

Před použitím služby je potřeba se zaregistrovat. Pro zprovoznění měření webu je nutné vytvořit nový profil webových stránek, ve kterém se vyplní URL adresa webu, který chceme sledovat. Následně systém vygeneruje měřicí kód, který je potřeba vložit do všech sledovaných stránek. Měřicí kód je napsán v jazyce JavaScript (viz ukázka kódu 2.2).

Google Analytics nabízí široké spektrum měřených dat. Kromě základních údajů, jako je počet zobrazení stránek, počet návštěv a počet návštěvníků, poskytuje i další informace o chování návštěvníků na webu — např. míru opuštění, průměrnou dobu, kterou uživatel na webu strávil, věrnost návštěvníků (jak často se na web vrací), nebo hloubku návštěvy uživatelů (kolik stránek uživatel zobrazil během jedné návštěvy).

Důležitá jsou data o zdrojích návštěvnosti, na která lze nahlížet v souvislosti s dosahováním cílů a konverzním poměrem. Např. pokud sledujeme e-shop, je možné nastavit Google Analytics tak, aby zaznamenával provedené objednávky a z nich tvořil zajímavé statistiky. Díky nim můžeme zjistit, jaký konverzní poměr vykazují např. návštěvníci z určitého vyhledávače, nebo placené reklamní kampaně, navíc u každého zdroje můžeme vidět i relevantními návštěvníky vygenerovaný obrat.

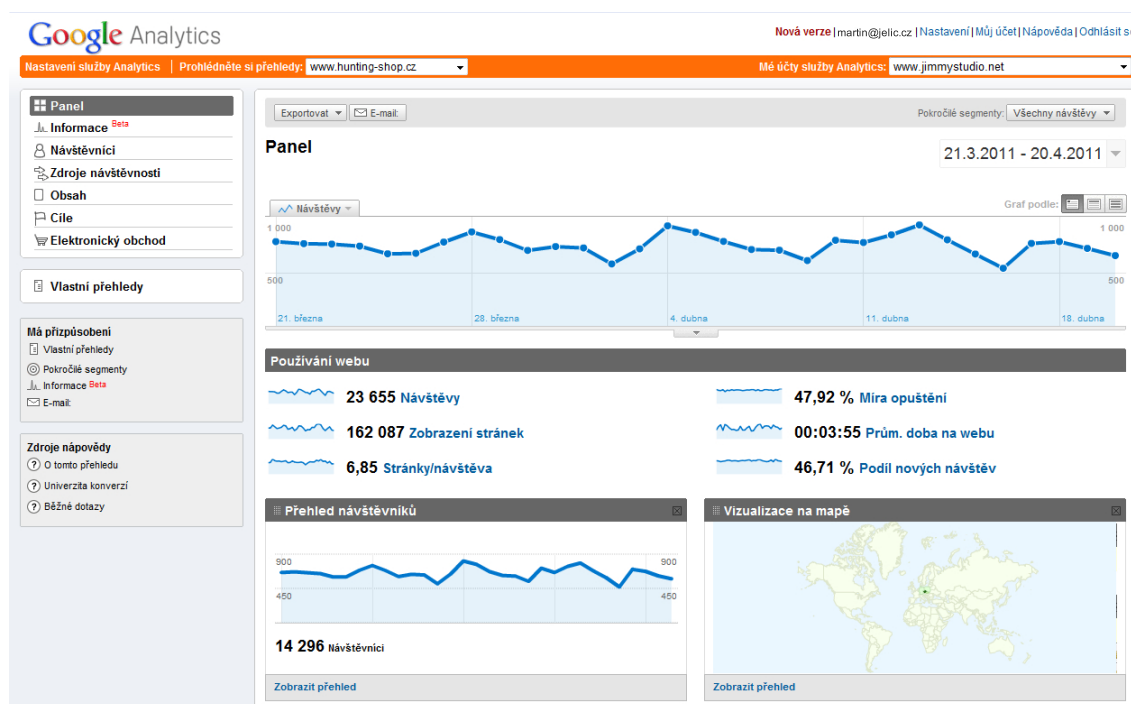
Součástí Google Analytics je i geografická vizualizace návštěv — můžeme tak zjistit, z jakých míst na Zemi lidé nejčastěji web navštěvují. Rozlišení funguje až na úroveň měst.

### 3.2 Yahoo Web Analytics

Pokročilý nástroj pro webovou analytiku, který je často srovnáván s Google Analytics, přestože cílová skupina těchto dvou nástrojů se výrazně liší. Zatímco Google se snaží svým nástrojem cílit na co největší spektrum uživatelů od odborníků až k úplným laikům (a to s sebou nese řadu kompromisů), Yahoo se soustředí pouze na odborníky a větší projekty. Stejně jako u Google se jedná o hostovanou službu. Samotné použití Yahoo Web Analytics je zdarma, avšak registrace je možná pouze prostřednictvím firem, které jsou členem Yahoo

---

<sup>1</sup>SaaS — Software as a service — tímto pojmem se označuje software, který je poskytován koncovým zákazníkům ve formě služby, někdy se pro tento model používá též pojem „cloud computing“. Samotný software je umístěn na serveru provozovatele a zákazník jej používá on-line, bez instalace na lokální počítač.



Obrázek 3.1: Google Analytics — prezentace naměřených dat — přehled

Web Analytics Consultant Network. V České republice je takových firem několik (viz sekce 3.9). Narozdíl od Google Analytics se tedy u Yahoo nejedná o „nástroj pro masy“ a tomu odpovídá také tržní podíl, který je oproti Google Analytics mizivý [12]. Nástroj od Yahoo používají spíše velké firmy, které využívají služeb poradenských firem. V ČR používá tento nástroj např. státní společnost České dráhy.

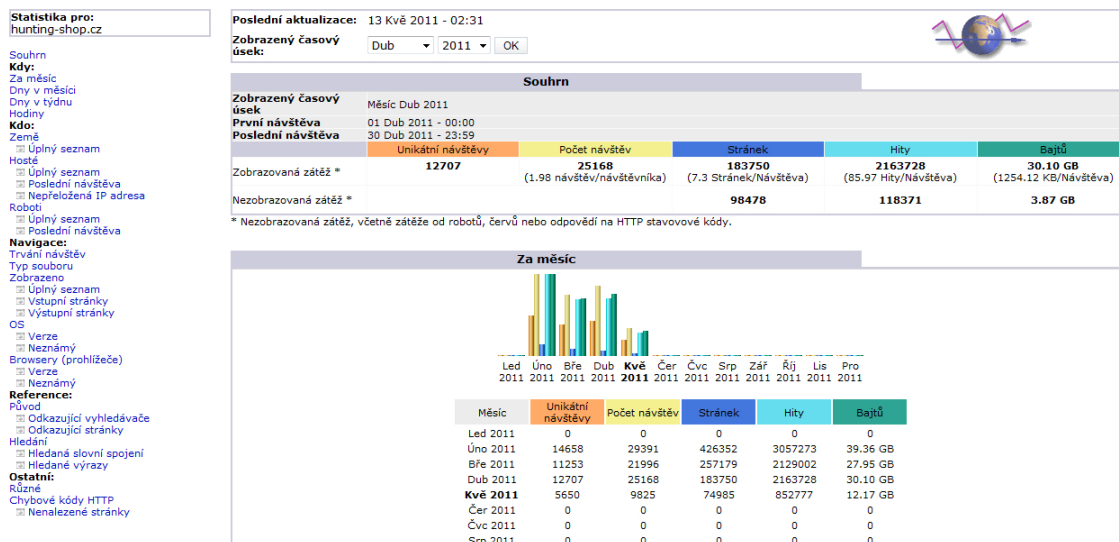
Funkčnost a možnosti Yahoo Web Analytics odpovídají zvolené cílové skupině. Nástroj od Yahoo má oproti Google Analytics mnohem větší možnosti přizpůsobení napříč celým systémem. Lze poměrně jednoduše vytvářet vlastní reporty dat, provádět různé křížové filtrace, segmentace návštěvníků podle libovolných parametrů apod. [18] V rukou odborníka se tedy jedná o velmi silný nástroj.

### 3.3 Adobe SiteCatalyst

Adobe SiteCatalyst (dříve Omniture SiteCatalyst<sup>2</sup>) je zástupcem nejvyšší kategorie nástrojů pro webovou analytiku. Jedná se o jeden z nejpokročilejších a také nejdražších softwarových produktů. Podobně jako u Google Analytics a Yahoo Web Analytics se i zde jedná o hostovanou službu. Ceny za použití se pohybují v řádu tisíců EUR měsíčně [17].

Oproti výše zmíněným nástrojům poskytuje SiteCatalyst rozšířené možnosti v oblasti přizpůsobení (systém lze kompletně nastavit dle individuálních potřeb, vč. struktury menu, reportů, grafů), lze definovat vlastní metriky, apod. Dále SiteCatalyst umožňuje integraci

<sup>2</sup>Nástroj SiteCatalyst byl vyvinut společností Omniture, která byla v roce 2009 koupena společností Adobe. V současné době jsou produkty Omniture zařazeny do balíku Adobe Online Marketing Suite.



Obrázek 3.2: AWStats — obrazovka se souhrnným přehledem naměřených údajů

s různými firemními systémy (BI<sup>3</sup>, CRM<sup>4</sup>) a umožňuje data v systému analyzovat až na úrovni jednotlivého návštěvníka.

Konkurencí pro SiteCatalyst jsou např. nástroje Coremetrics Analytics, Unica NetSight, nebo Webtrends Analytics. Popisem těchto nástrojů se z prostorových důvodů tato práce nezabývá.

### 3.4 AWStats

Jedná se o zástupce nástrojů, založených na analýze logu webového serveru. První verze byla vydána v roce 2000 a nástroj se díky své přehlednosti, jednoduchosti a ceně (je zdarma, šířen pod licencí GPL<sup>5</sup>) rychle rozšířil. Mnoho poskytovatelů webhostingů nabízí AWStats v ceně hostingového programu. Jak vypadá prostředí software AWStats nastiňuje obrázek 3.2.

### 3.5 Webalizer

Další zástupce nástrojů, které analyzují log soubor webového serveru. Vznikl v roce 1997 a jedná se o velice primitivní nástroj, v praxi pro webovou analytiku téměř nepoužitelný. Největším problémem softwaru Webalizer je neschopnost odlišit přístupy robotů od přístupů lidských. To způsobuje významné zkreslení výsledků, někdy až o stovky procent. Další nevýhodou je ignorace části URL, označované jako query string<sup>6</sup>, kvůli čemuž jsou dvě

<sup>3</sup>BI — Business Intelligence — touto zkratkou se označuje software určený pro podporu řízení podnikání (někdy se používá též pojem „systémy pro podporu rozhodování“)

<sup>4</sup>CRM — Customer Relationship Management — touto zkratkou se označuje software určený pro podporu řízení vztahů se zákazníky

<sup>5</sup>GNU General Public License — licence pro svobodný software s otevřeným zdrojovým kódem. Vyžaduje, aby odvozená díla byla dostupná pod toutéž licencí.

<sup>6</sup>Jako query string je označována část URL za otazníkem, jež může obsahovat parametry, které jsou předávány webovému serveru.

stránky, liší se pouze parametrem v URL (např. *www.mujiweb.cz/index.php?page=znamky* a *www.mujiweb.cz/index.php?page=vysledky*), počítány dohromady jako jedna stránka.

## 3.6 TopList.cz

Jednoduchá česká služba, porovnávající vzájemně návštěvnost webů. Z naměřených hodnot sestavuje žebříčky návštěvnosti a souhrnné („globální“) statistiky. Jedná se o bezplatnou hostovanou službu, která funguje na principu měřicího kódu ve stránce. Měřicí kód využívá buď čistě HTML tag `<img>`, nebo jeho kombinaci s JavaScriptem (viz ukázky kódu 2.3 a 2.4).

TopList měří v základní konfiguraci pouze následující údaje: počet návštěv, počet zhlédnutí a počet unikátních IP. Rozšířená konfigurace umožňuje měřit ještě několik dalších údajů: zdroje návštěv (pouze ve formě výpisu jednotlivých celých URL, nerozpoznává vyhledávače, neparsuje klíčová slova, neseskupuje domény, neuvádí přímou návštěvnost), nejčastější vstupní stránky, informace o návštěvníkově vybavení (typ prohlížeče, operační systém, rozlišení obrazovky, barevná hloubka) apod.

TopList vzhledem ke svojí jednoduchosti nelze používat pro hlubší analýzu návštěvnosti, avšak užitečná je jeho srovnávací funkce. Je to vhodný doplněk k pokročilejším analytickým nástrojům (např. Google Analytics). Např. pokud v analytickém nástroji zjistíme náhlý významný pokles návštěvnosti, můžeme si v TopListu ověřit, zda obdobný pokles nepostihl i konkurenci (návštěvnost webů obecně kolísá na základě různých vnějších vlivů, jako je sezónnost, počasí, politická situace apod.).

## 3.7 mYx

Placená hostovaná služba mYx české firmy SiteOne je vhodným doplňkem k výše představeným analytickým nástrojům. Umožňuje vizualizovat pozice kliknutí návštěvníků ve formě tzv. heatmapy<sup>7</sup>. Heatmapa je v tomto případě poloprůhledná vrstva, překrývající samotnou webovou stránku. V této vrstvě jsou barevně znázorněny oblasti, ve kterých došlo ke kliknutí — čím více kliknutí je v dané oblasti zaznamenáno, tím „teplejší“ barvu oblast má (barevné spektrum se pohybuje od modré směrem ke žluté a červené).

Díky této vizualizaci lze snadno diagnostikovat chyby v použitelnosti stránek a optimalizovat jejich konverzní poměr. Zjistíme-li například, že většina uživatelů kliká do místa, které není aktivní (neobsahuje odkaz ani spuštění nějaké funkce), jedná se evidentně o chybu. Z vizualizace si lze též udělat částečnou představu o chování uživatelů na webu.

## 3.8 CrazyEgg

CrazyEgg je podobným nástrojem, jako výše představený mYx, avšak v analytických schopnostech jde o poznání dál. Kromě heatmapy umožňuje uživatelská kliknutí vizualizovat i jinými způsoby, pomocí nichž lze kliknutí dávat do souvislosti např. se zdroji návštěv, klíčovými slovy, odkazy na stránky, či s typy prohlížečů.

Jedná se o placenou hostovanou službu.

---

<sup>7</sup>Vychází z anglického pojmu „heat map“, kterým se označuje grafická reprezentace dat, ve které jsou hodnoty proměnné v dvojrozměrné tabulce reprezentovány pomocí barev.



## Kapitola 4

# Knihovna/nástroj pro sledování zdrojů a pohybu návštěvníků webu

### 4.1 Specifikace požadavků

Knihovna by měla umožňovat sledování návštěvníků webových stránek. Konkrétně by mělo být možné sledovat, z jakých zdrojů návštěvníci přicházejí (přímá návštěvnost, vyhledávače, odkazující weby, reklamní kampaně) a jakým způsobem se v rámci webu pohybují (jaké stránky webu postupně navštívili a kde své putování ukončili). Knihovna by uvedené úkoly měla řešit kompletně, tedy včetně sběru a ukládání dat a jejich následné analýzy. Přístup k datům by měl být umožněn prostřednictvím API, které vrací data v některém ze stanardních formátů. Zároveň by měla být nad knihovnou postavena prezentační vrstva, která by umožnila přímý přístup k výsledkům měření ve srozumitelné formě. Díky tomu by mělo jít knihovnu použít jako samostatný nástroj pro webovou analytiku.

### 4.2 Název knihovny/nástroje

Aby bylo možné na knihovnu v textu snadno odkazovat, pojmenoval jsem projekt názvem **MJmonitor**. Název vychází z iniciálů autora a slova „monitoring“. Kromě tohoto názvu je v následujícím textu použita i zkratka **MJM**.

### 4.3 Použité technologie

Již v zadání práce je dáno, že knihovna má být implementována v jazyce PHP. Tento jazyk je natolik známý a rozšířený, že není potřeba se o něm zde podrobněji rozepisovat. Totéž platí o dalších použitých jazycích: JavaScript, XHTML, CSS. Proto v této kapitole zmíním pouze dvě méně známé technologie, které jsou v projektu použity.

#### 4.3.1 Nette framework

Nette framework je open-source knihovna v jazyce PHP, která má za úkol usnadnit programátorovi tvorbu webových aplikací, eliminovat bezpečnostní rizika a také do určité míry standartizovat architekturu aplikací. Knihovna je šířena pod licencemi New BSD<sup>1</sup> a GNU

---

<sup>1</sup>New BSD — jedna z nejsvobodnějších licencí pro svobodný software



GPL<sup>2</sup>. Původním autorem knihovny je David Grudl<sup>3</sup>, v současné době se na vývoji frameworku podílí početná komunita vývojářů.

Základem aplikací postavených nad Nette je architektura MVC<sup>4</sup>, která rozděluje aplikaci na tři vrstvy — model zajišťuje práci s daty, pohled (View) zajišťuje prezentaci dat a řadič (Controller, v případě Nette frameworku se používá pojem Presenter) řídí běh aplikace — reaguje na události a zajišťuje změny v modelech a pohledech.

Nette framework disponuje mimo jiné vlastním šablonovacím systémem, který se automaticky stará o eliminaci bezpečnostních děr typu XSS<sup>5</sup>, CSRF<sup>6</sup> a dalších. Velkou výhodou frameworku jsou též kvalitní ladící nástroje.

### 4.3.2 MongoDB

Jedná se o bezschémový, dokumentově orientovaný databázový systém, který pracuje s kolekcemi dokumentů ve formátu JSON<sup>7</sup>. Systém je napsán v jazyce C++ a je šířen pod open-source licencí GNU AGPL v3.0. O vývoj systému se stará americká společnost 10gen.

Volba tohoto typu databázového systému může být pro některé čtenáře překvapivá. Nabízí se použití některého z relačních databázových systémů (MySQL, PostgreSQL, apod.), které jsou známější a rozšířenější. Bezpochyby by se tento typ databáze dal pro účely popisovaného projektu použít (důkazem budiž česká služba TOPlist, která využívá MySQL). Nicméně několik důvodů vedlo k finálnímu rozhodnutí použít pro tento projekt MongoDB.

Nelze popřít, že jedním z důvodů byla vlastní zvědavost, zájem vyzkoušet si méně rozšířený databázový systém a zjistit, jak obstojí v aplikaci typu OLAP. Kromě subjektivních zájmů bylo však rozhodnutí podloženo i několika výhodami a referencemi z oblasti analytických systémů.

Výhodou dokumentově orientované databáze může být v případě tohoto projektu její bezschémovost. Představme si, že sesbíraná data ukládáme do tabulky relační databáze, přičemž jedno zobrazení stránky odpovídá jednomu řádku v tabulce. Časem bude tabulka obsahovat statisíce až miliony řádků a pokud budeme chtít změnit strukturu ukládaných dat, změna schématu tabulky bude velice časově a výpočetně náročnou operací. Oproti tomu u dokumentově orientované databáze jednoduše začneme ukládat do stejné kolekce data v nové struktuře a zároveň odpovídajícím způsobem upravíme analytické algoritmy, přičemž můžeme zachovat zpětnou kompatibilitu s původní strukturou dat.

Další výhodou, bavíme-li se konkrétně o MongoDB, je rychlost. Vzhledem k absenci schémat, relací, transakcí a dalších principů relačních databází (bez kterých se tento projekt obejde) je systém jednodušší, a to se projevuje na jeho rychlosti. Nad rychlostí by se samozřejmě daly vést různé diskuse — jistě by se našly i případy, ve kterých by relační databáze byla rychlejší. Nicméně na internetu se dá nalézt několik benchmarků (viz [19], [15] a [16]), ze kterých je patrné, že např. oproti MySQL je MongoDB často i několikrát rychlejší, a to jak ve vkládání záznamů, tak v provádění dotazů nad kolekcemi.

MongoDB je rovněž velmi dobře škálovatelným databázovým systémem. V případě, že na databázové operace nestačí výkon jednoho serveru (což se při monitorování většího množ-

---

<sup>2</sup>GNU General Public License — licence pro svobodný software. Vyžaduje, aby odvozená díla byla dostupná pod toutéž licencí.

<sup>3</sup>David Grudl je český PHP programátor, kromě frameworku Nette je i autorem open-source knihoven Taxy! a Dibi.

<sup>4</sup>MVC — Model-View-Controller

<sup>5</sup>XSS — Cross Site Scripting

<sup>6</sup>CSRF — Cross Site Request Forgery

<sup>7</sup>JavaScript Object Notation - jednoduchý formát pro výměnu dat

ství webů s vysokou návštěvností může stát), MongoDB disponuje relativně jednoduchými a efektivními prostředky pro replikaci dat na více serverů.

Co se týče referencí, jsou známy úspěšné aplikace MongoDB pro účely webových analytických systémů. Příkladem budiž americká společnost Gilt Groupe, provozující zvláštní internetový obchod se zbožím luxusních značek. Tato společnost využívá MongoDB pro sběr a analýzu dat o chování uživatelů na webu [4] a zároveň nad touto databází postavila nástroj Hummingbird[1], jenž je určen pro sledování provozu na webu v reálném čase.

Pro PHP existuje rozšíření, které umožňuje v tomto jazyce pracovat s databází MongoDB. Toto rozšíření je poměrně kvalitně zdokumentováno v oficiálním manuálu PHP [13]. V projektu je použito pro veškerou práci s databází.

## 4.4 Návrh

V rámci knihovny je potřeba řešit tři základní fáze činnosti:

1. Sběr dat
2. Zpracování a analýza dat
3. Prezentace výsledků

Ve všech třech fázích pracujeme s daty, a tato data je potřeba někam ukládat. Na jedné straně bude probíhat sběr dat, na straně druhé se budou tato data analyzovat a prezentovat. Nasbíraná data se nebudou nijak měnit, budou uchovávána historicky a budou označena časem pořízení. Výstupy ze systému jsou určeny pro podporu rozhodování v oblasti internetového marketingu. Z uvedených informací je zřejmé, že se jedná o systém typu OLAP<sup>8</sup> a je tedy vhodné se některými principy této technologie inspirovat.

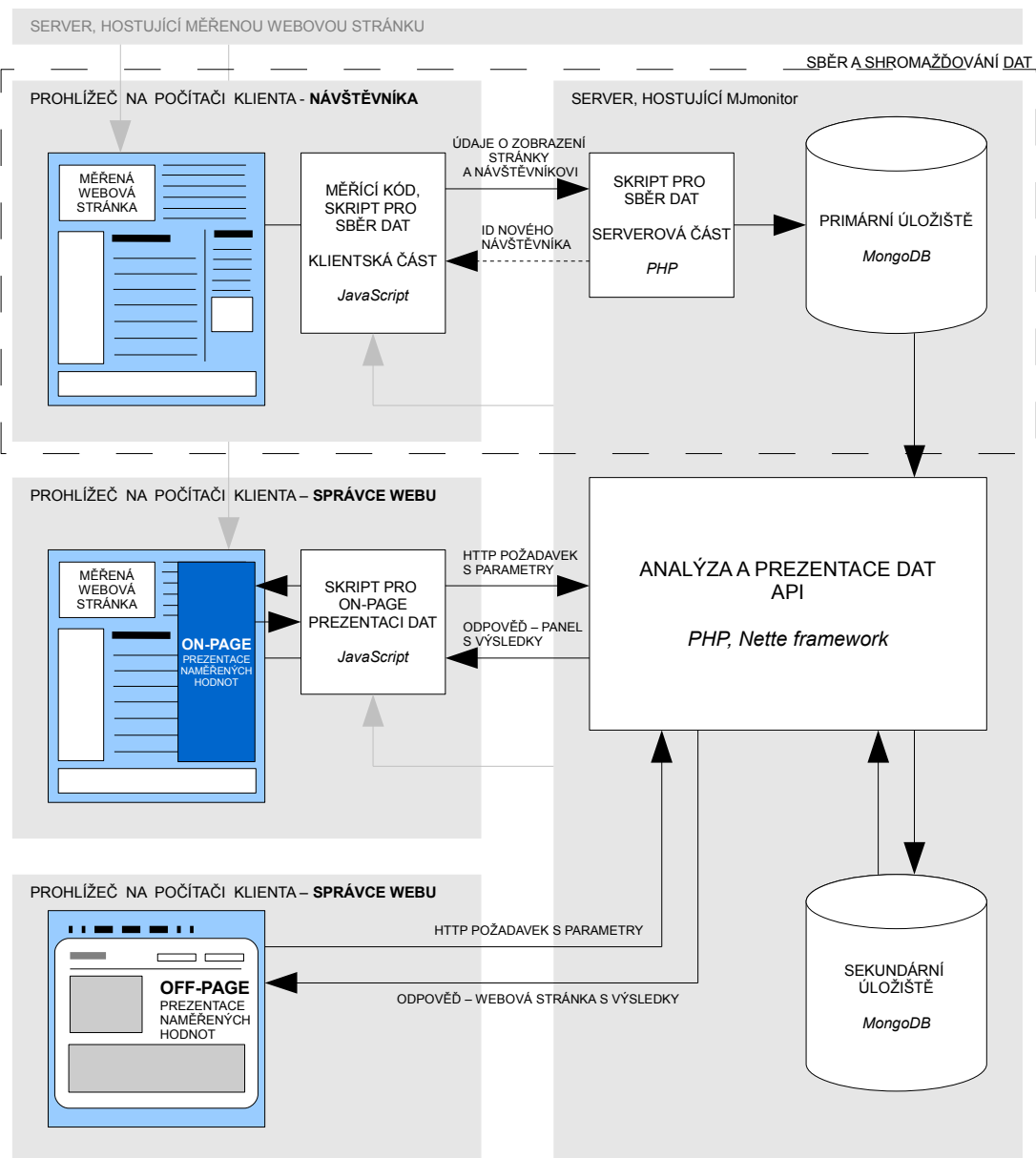
MJmonitor je navržen tak, aby se dal použít pro měření několika různých webů. Nemusí být nainstalován na stejném serveru, na kterém běží měřený web. Naopak je možné jej nainstalovat na jeden server a měřit pomocí něho weby na několika jiných serverech.

Navržená architektura je znázorněna na obrázku 4.1. O sběr dat se stará na straně klienta měřicí kód ve stránce a skript pro sběr dat (ten implementuje potřebnou funkčnost). Sesbíraná data jsou odeslána serverovému skriptu. Ten má na starosti ošetření přijatých dat a jejich uložení do primárního (sběrného) úložiště, případně vygenerování identifikačního řetězce pro nového návštěvníka. Zpracování, analýzu a prezentaci dat má na starosti jádro knihovny. To pracuje s daty v primárním úložišti, která analyzuje a výsledky analýz buďto přímo prezentuje, nebo ukládá do sekundárního úložiště, odkud je možné je později vyvolat, případně na nich provádět další jednodušší analýzy a agregační operace. Prezentace dat probíhá dvěma způsoby. Přímou na měřeném webu je možné prezentovat výsledky, týkající se aktuálně zobrazené stránky (on-page), nebo lze prezentovat souhrnné výsledky, týkající se celého webu, prostřednictvím zvláštního webového rozhraní (off-page). Je k dispozici též jednoduché API, které umožňuje pracovat s výsledky měření např. v administračním rozhraní CMS<sup>9</sup>. V následujících podkapitolách postupně probereme návrh jednotlivých částí systému.

<sup>8</sup>OLAP — On-Line Analytical Processing

<sup>9</sup>CMS — Content Management System — systém pro správu obsahu webových stránek, často se používá též nepřesný pojem „redakční systém“.





Obrázek 4.1: Blokové schéma, znázorňující architekturu celého systému

#### 4.4.1 Sběr dat

Jak bylo zmíněno v kapitole 2.2, základní možnosti pro sběr dat jsou dvě:

1. Logování přístupů na serveru
2. Zasílání dat na server pomocí měřicího kódu ve stránce

První způsob probíhá většinou automaticky v rámci webového serveru. Druhý způsob vyžaduje vložení měřicího kódu do všech stránek webu, které chceme měřit. Vzhledem k výhodám, popsaným v sekci 2.2.2, byl pro účely tohoto projektu zvolen způsob druhý, tedy měřicí kód ve stránce.

V počáteční fázi návrhu projektu bylo potřeba se zamyslet nad zásadní otázkou, jaká data sbírat a ukládat. Měřicí kód bude umístěn ve všech měřených stránkách webu, informace budeme tedy předávat serveru při každém zobrazení stránky. Protože každé zobrazení stránky je jednorázová akce a není možné získat jakékoli informace dodatečně, je vhodné dobře promyslet strukturu dat, která budeme sbírat, s ohledem na to, jaké výstupy ze systému budeme potřebovat (i v případných budoucích verzích). Následuje výčet údajů, které budeme sbírat, včetně vysvětlení, k čemu je daný údaj užitečný.

**URL stránky, kterou uživatel právě zobrazil** Údaj je důležitý pro sledování pohybu návštěvníka v rámci webu.

**Titulek stránky, kterou uživatel právě zobrazil** Jedná se o doplňující údaj. Pro sledování pohybu návštěvníka v rámci webu by stačil údaj o URL adrese stránky, nicméně pro uživatelsky přívětivou prezentaci výsledků je vhodné znát i titulek stránky, neboť ten zpravidla lidsky srozumitelnější formou popisuje, o jakou stránku se jedná.

**URL odkazující stránky** Jedná se o adresu stránky, z níž návštěvník na právě zobrazenou stránku přišel prostřednictvím hypertextového odkazu. Pokud tento údaj nebude k dispozici, můžeme usuzovat, že návštěvník na stránku přišel přímo — tedy zadáním URL do adresního řádku prohlížeče, případně prostřednictvím odkazu v záložkách, anebo prostřednictvím hypertextového odkazu v dokumentu, který nelze adresovat pomocí URL (e-mail, off-line PDF dokument, apod.).

**Unikátní identifikátor návštěvníka** Každému<sup>10</sup> návštěvníkovi bude v rámci systému při prvním zobrazení stránky přidělen unikátní kód, pomocí kterého bude rozpoznán při dalších přístupech na web. Tento údaj je velmi důležitý a využití nalezne napříč celým systémem — např. při analýze počtu návštěv, počtu unikátních návštěvníků, průměrného času zobrazení stránky, pohybu jednotlivých návštěvníků v rámci webu, apod.

**Razítko prohlížeče** Jedná se o informaci o typu a verzi internetového prohlížeče, pomocí kterého bylo zobrazení stránky provedeno. Na základě tohoto údaje lze zjistit např. procentuální rozložení návštěvníků webu dle typů a verzí prohlížeče. Tento údaj je důležitý pro rozhodování o podpoře jednotlivých prohlížečů. Např. pokud zjistíme, že starý prohlížeč Internet Explorer verze 6, který je z hlediska vývoje webových stránek velmi problémový, používá méně než desetina procenta uživatelů, můžeme usoudit, že podpora tohoto prohlížeče se nevyplatí.

---

<sup>10</sup>Slovo „každému“ zde neplatí stoprocentně. Více v kapitole 4.5.1

**IP adresa / hostname klientského počítače** Na základě tohoto údaje lze orientačně zjišťovat geografickou polohu návštěvníků. Údaj lze rovněž využít ke kontrole identifikace návštěvníka.

**Časové razítko** Je důležité vědět, kdy se zobrazení stránky událo. Tento údaj je využitelný napříč celým systémem. Veškeré výsledky lze díky němu omezovat určitým časovým intervalem.

#### 4.4.2 Ukládání dat

Výše uvedené údaje, vycházející ze sběrné části projektu, je potřeba někde shromažďovat. Vzhledem k charakteru systému je nutné předpokládat, že nad sebranými údaji se budou provádět různé typy analytických operací, od triviálních agregací až po složité vícefázové analýzy. Výběr úložiště rozebírá kapitola 4.3.

Pro ukládání sbíraných dat postačí jediná kolekce, kterou pojmenujeme *pageViews*. Do této kolekce bude při každém zobrazení stránky uložena jedna položka, obsahující výše uvedené údaje. Analýzou údajů v této kolekci můžeme získat všechny potřebné výsledky.

#### 4.4.3 Analýza dat

Nasbíraná „hrubá“ data v kolekci *pageViews* je potřeba různými způsoby analyzovat, abychom z nich získali užitečné výsledky.

Co je potřeba analyzovat:

- údaje o návštěvách — počet návštěv, počet zobrazení stránek, průměrná délka návštěvy, míra odrazu
- údaje o zdrojích návštěvnosti — počty přístupů z jednotlivých zdrojů, pokud je zdrojem vyhledávač, zjistit četnost hledaných výrazů
- údaje o návštěvnicích a jejich chování — typ prohlížeče, navštívené stránky v rámci webu

Protože záznamů v kolekci *pageViews* může být obrovské množství, mohou mít některé složitější analytické operace neúnosně vysokou časovou náročnost. Tento problém lze řešit rozdělením analýzy do dvou, nebo více fází. Např. lze v denních intervalech automatizovaně (např. pomocí CRONu<sup>11</sup>) provádět analýzu údajů o návštěvách a pro každý den uložit do zvláštní kolekce záznam s agregovanými údaji. Získání souhrnných údajů pro určitý časový interval (např. týden) potom znamená pouze vhodným způsobem sečíst údaje za jednotlivé dny.

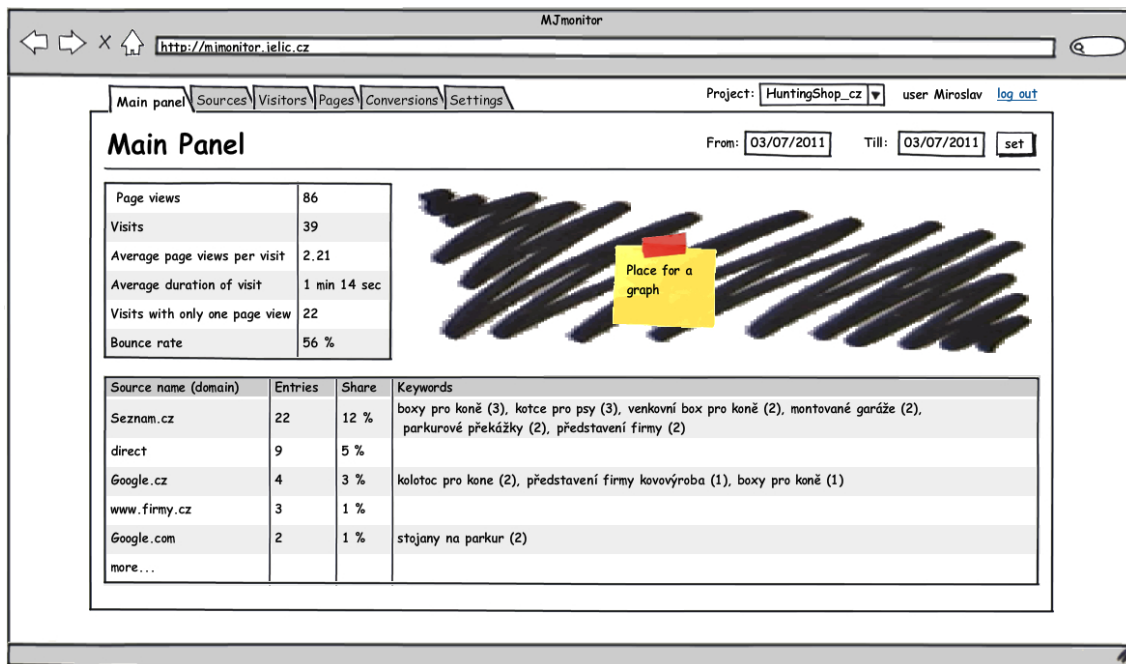
#### 4.4.4 Presentace výsledků

Výsledky, získané analýzou nasbíraných údajů, je potřeba nějakým způsobem prezentovat. Knihovna umožňuje prezentovat data ve formě srozumitelné člověku i počítačovému programu (API). Celý systém je vyveden v anglickém jazyce. Prezentační část je rozdělena tří částí:

##### 1. Off-page prezentace souhrnných výsledků

---

<sup>11</sup>CRON je služba, která v operačních systémech automatizovaně spouští v určitý čas nějaký příkaz.



Obrázek 4.2: Skica uživatelského rozhraní off-page části

2. On-page prezentace výsledků, které se váží k jednotlivým stránkám
3. API, poskytující údaje ve formátu JSON

## Off-page část

Off-page část má za úkol prezentovat souhrnné výsledky pro celý sledovaný web v lidsky srozumitelné formě. Do systému je potřeba se přihlásit za pomoci uživatelského jména a hesla. Uživatelské rozhraní systému je navrženo v minimalistickém stylu, cílem je maximální přehlednost a orientace na data. Náčrt uživatelského rozhraní je možno vidět na obrázku 4.2.

Napříč všemi sekcemi systému by se mělo nacházet záložkové menu, dále rozbalovací seznam pro výběr webu, s jehož výsledky chceme pracovat, jméno přihlášeného uživatele, odkaz pro odhlášení a lišta s formulářem pro časový interval sledování (datum od-do).

Na úvodní stránce by se měl nacházet stručný souhrn naměřených výsledků všech typů. V sekci Sources by měl být podrobnější rozbor zdrojů návštěvnosti — četnost přístupů z jednotlivých zdrojů, případně klíčová slova. Sekce Visitors by měla poskytovat údaje o jednotlivých návštěvnících — zejména o jejich pohybu v rámci webu.

## On-page část

On-page část je velmi přímočará a rychlá metoda prezentace výsledků, týkajících se jednotlivých stránek webu. Jedná se o rámeček, který lze zobrazit přímo na sledovaném webu. V rámečku by se měly zobrazovat naměřené údaje pro aktuálně zobrazenou stránku — počet zobrazení stránky, počet odchodů, průměrný čas strávený na stránce, odkud návštěvníci na stránku nejčastěji přicházejí a kam nejčastěji odchází.

## API

API by mělo poskytovat jednoduché rozhraní, které umožní jiným programům přístup ke všem prezentovatelným výsledkům měření.

## 4.5 Implementace

Jak již bylo zmíněno v kapitole 4.3, systém je implementován v kombinaci jazyků PHP, JavaScript, XHTML a CSS.

### 4.5.1 Sběr a ukládání dat

Ke sběru dat u klienta slouží soubor *tracker.js*, který, jak napovídá koncovka, je napsán v jazyce JavaScript a obsahuje funkce pro zjištění potřebných údajů, identifikaci návštěvníka a odeslání získaných informací na server. V zájmu minimalizace možných kolizí s jinými JavaScriptovými knihovnami jsou všechny funkce implementovány jako metody objektu *mjm*.

K tomu, aby sběr dat začal fungovat, je potřeba vložit do sledovaného webu měřicí kód. Ten je napsán v jazyce JavaScript, jehož kód je obalen do odpovídajících XHTML značek. V praxi vypadá měřicí kód takto:

```
<script type="text/javascript"
  src="http://umisteni.knihovny/tracker.js"></script>
<script type="text/javascript">
  try {
    mjm.setUrl('http://umisteni.knihovny');
    mjm.setProject('NazevProjektu');
    mjm.trackPageView();
  } catch (err) {}
</script>
```

Kód 4.1: Příklad měřicího kódu

V prvním bloku `<script>` je do stránky vložen zdrojový kód souboru *tracker.js*, který obsahuje potřebnou funkčnost. V atributu *src* je potřeba nastavit správnou URL adresu souboru. Ta závisí na umístění knihovny. Druhý blok obsahuje kód, který již pracuje s metodami objektu *mjm* ze souboru *tracker.js*. Nejprve je potřeba nastavit sběrný skript. Metodě *mjm.setUrl()* předáváme jako parametr URL adresu knihovny, dále pomocí metody *mjm.setProject()* nastavíme název projektu (tento název slouží jako identifikace měřeného webu) a nakonec zavoláme metodu *mjm.trackPageView()*, která zajistí samotné zaznamenání zobrazení stránky knihovnou. Kód je uzavřen do bloku *try...catch*, který zajišťuje, aby případné chyby, vzniklé při provádění funkcí, byly ignorovány a návštěvník webu je nezaznamenal.

Metoda *mjm.trackPageView()* nejprve zjistí potřebné údaje. URL adresa<sup>12</sup> zobrazené stránky je konkaténací atributů *window.location.pathname* a *window.location.search*, titulek stránky lze zjistit z proměnné *document.title* a URL adresu odkazující stránky poskytuje atribut *document.referrer*. Dále metoda zjistí, zda lze návštěvníka identifikovat, nebo se jedná o jeho první přístup na web. Pokud návštěvník na webu již byl, měla by existovat

<sup>12</sup>Nejedná se o plnohodnotnou URL adresu. Pro účely monitorování návštěvnosti webu stačí část URL za doménou 1. řádu - tedy např. pro URL adresu `http://www.hunting-shop.cz/produkty/bunda-fleece-softshell/107015001/` se uloží pouze část `/produkty/bunda-fleece-softshell/107015001/`. Doménové jméno je pro všechny stránky v rámci měřeného webu stejné a je uloženo v nastavení projektu.

cookie *mjmVisitorID*, obsahující identifikační řetězec návštěvníka. V takovém případě je řetězec odeslán na server spolu s ostatními daty. Pokud zmíněná cookie neexistuje, znamená to, že se jedná buď o nového návštěvníka, nebo o návštěvníka, jehož prohlížeč nepodporuje cookies. Aby nedocházelo ke zkreslení výsledků uživateli nepodporujícími cookies, je u každého „nového“ návštěvníka proveden test podpory cookies. Pokud tento test dopadne negativně, návštěvník je ignorován (jeho zobrazení stránky se do systému vůbec nezaznamená). V opačném případě jsou data o zobrazení stránky odeslána na server bez identifikačního řetězce. Server data přijme, vygeneruje nový identifikační řetězec a metodou JSON-P jej zašle zpět klientskému skriptu, který jej uloží do cookie.

Na straně serveru je implementace velmi jednoduchá. Skript *tracker.php* má na starost přijetí dat od klientského skriptu standardní metodou GET a uložení těchto dat do kolekce *pageViews* v databázi (příklad záznamu v kolekci *pageViews* prezentuje ukázka 4.2). Nutno podotknout, že klientský skript zasílá pouze část potřebných údajů, zbylé údaje zjišťuje serverový skript sám — jedná se o razítko prohlížeče (v PHP proměnná `$_SERVER['USER_AGENT']`), dále IP adresu (`$_SERVER['REMOTE_ADDR']`) a časové razítko (funkce `time()`). V případě, že se jedná o nového návštěvníka, musí skript vygenerovat identifikační řetězec a poslat jej metodou JSON-P zpět klientskému skriptu, aby mohl být uložen u návštěvníka do cookie. Pokud se jedná o opakovaný přístup již identifikovaného návštěvníka, serverový skript obdrží identifikační řetězec spolu s ostatními údaji od klientského skriptu.

```
{
  "_id": ObjectId("4dce926f00b55e844c000000"),
  "pageUrl": "/sekce/493-helle-cepele/",
  "pageTitle": "Helle_čepele_|_Helle_čepele,_čepel_Helle,_třívrstvý_
    sendvič,_norské_čepele,_nerezová_čepel,_karbonová_čepel_|_
    Hunting-shop",
  "referrerUrl":
    "http://hunting-shop.cz/produkty/brusletto-cepel-veskeblad/BRU20600/",
  "userAgent": "Mozilla/4.0_(compatible;_MSIE_7.0;_Windows_NT_6.0;_GTB6.6;_
    SLCC1;_.NET CLR_2.0.50727;_.NET CLR_3.5.30729;_.NET CLR_3.0.30729)",
  "ip": "213.226.224.118",
  "host": "exchange.vpsji.cz",
  "time": 1305383535,
  "visitorID": "4dce8f901922a"
}
```

Kód 4.2: Příklad záznamu o zobrazení stránky v kolekci *pageViews* (data jsou vypsána ve formátu JSON)

## 4.5.2 Analýza dat

Analýzu dat implementují statické metody třídy *Analyzer*. Podkladem veškerých analýz jsou data v kolekci *pageViews*.

Jednoduchý výběr dat z databáze MongoDB probíhá výběrem kolekce a následným položením dotazu, kterému se jako jeden argument předá pole podmínek a jako druhý argument pole klíčů, které chceme ve výsledku vrátit. Oba argumenty jsou volitelné. Přístup je podobný, jako u SQL výrazu „*SELECT [...] FROM [...] WHERE [...]*“, pouze zápis je zcela jiný. Příklad kódu v jazyce PHP si lze prohlédnout v ukázce 4.3.

```

// v proměnné $db máme instanci databázové třídy
$collection = $db->pageViews;
$cursor = $collection->find(array('visitorID' => '4dcbb89c1efe3'),
    array('pageUrl'));
// nyní máme v proměnné $cursor iterátor, pomocí kterého můžeme
    vypsat výsledky
    foreach ($cursor as $item) {
        echo $item['pageUrl'];
    }

```

Kód 4.3: Ukázka práce s MongoDB v PHP — vypsaní URL adres všech stránek, které navštívil uživatel s identifikačním řetězcem *4dcbb89c1efe3*

Složitější agregační a analytické operace se v MongoDB implementují v jazyce JavaScript za použití programovacího modelu Map/Reduce [2]. Tento model byl vyvinut společností Google pro paralelní zpracování a generování velkých množin dat [7]. Princip je následující:

Definujeme funkci *map*, která bude z jednotlivých prvků množiny<sup>13</sup> generovat pracovní dvojice (*klíč, hodnota*), přičemž z jednoho prvku může funkce vygenerovat libovolný počet dvojic. Dále definujeme funkci *reduce*, která bude dostávat na vstup vždy klíč a sadu hodnot, vygenerovaných pro tento klíč funkcí *map*. Funkce *reduce* má za úkol tyto hodnoty nějakým způsobem spojit a vrátit jedinou hodnotu. Struktura hodnoty, kterou vrací funkce *reduce*, musí být stejná jako struktura hodnoty, kterou generuje funkce *map*. Funkce *reduce* totiž může být pro jeden klíč vykonána několikrát, spojování hodnot může probíhat postupně v jakési stromové hierarchii. V takovém případě je výstup z funkce *reduce* použit jako vstup do dalšího volání stejné funkce, proto musí být struktura vrácené hodnoty stejná, jako u funkce *map*.

Výstupy z procesu Map/Reduce nad určitou kolekcí jsou ukládány do jiné kolekce, jejíž název je buď pevně stanoven, nebo automaticky vygenerován (dočasná kolekce). MongoDB k modelu Map/Reduce doplňuje ještě volitelnou funkci *finalize*, která umožňuje provést konečné úpravy záznamů ve výstupní kolekci. Tato funkce dostává na vstup pro každou položku dvojici (*klíč, hodnota*) a vrací upravenou hodnotu. Funkce se dá použít např. pro změnu struktury záznamů, nebo pro řadící a agregační operace v rámci jednotlivých záznamů.

Pro ilustraci využití modelu Map/Reduce následuje popis jednoho z algoritmů, vytvořených při implementaci knihovny.

### Algoritmus pro výpočet údajů o návštěvách za urč. časový úsek

Pracujeme se sadou záznamů z kolekce *pageViews* omezenou zvoleným časovým intervalem. V tomto případě počítáme výsledky vždy za jeden den (od 0:00:00 do 23:59:59) a vypočítané hodnoty uložíme dohromady jako záznam do tabulky *visitsDailyTotals*. Ta potom obsahuje takovýto záznam pro každý den a výpočet souhrnných údajů za zvolený časový interval v délce několika dní je pouhou agregací výsledků za jednotlivé dny. Den je v aktuální verzi systému nejmenší časovou jednotkou, za kterou lze zjistit souhrnné výsledky.

**funkce *map*** Funkce *map* emituje pro každý záznam v kolekci *pageViews* klíč, odpovídající identifikačnímu řetězci návštěvníka a hodnotu ve tvaru *{počet zobrazení stránek: 1, seznam*

<sup>13</sup>Množinou se v tomto případě rozumí sada dokumentů vybrané kolekce. Sadu lze omezit použitím standardního vyhledávacího dotazu, který vybere pouze záznamy, vyhovující určitým podmínkám.



časů zobrazení: [časové razítko aktuálního záznamu]}.

**funkce *reduce*** Funkce *reduce* spojuje emitované záznamy na základě shody klíče, přičemž sčítá počet zobrazení stránek a kompletuje seznam časů zobrazení. Výstupem je tedy hodnota ve tvaru {počet zobrazení stránek: (součet pro jednoho návštěvníka), seznam časů zobrazení: [(časová razítka zobrazení, vykonaných jedním návštěvníkem)]}.

**funkce *finalize*** Funkce *finalize* je v rámci tohoto algoritmu nejsložitější. Nejprve seřadí časová razítka od nejstaršího k nejnovějšímu a dále seznam těchto časů prochází v cyklu, přičemž zjišťuje počet návštěv, počet singletonů (viz sekce 2.1.6) a celkový čas, který návštěvník na webu strávil. Jednotlivé návštěvy jsou odděleny časovou prodlevou minimálně 30 minut mezi zobrazeními stránek. Pokud tedy cyklus narazí na časové razítko, lišící se od předchozího o více než 30 minut, započítá novou návštěvu a zároveň započítá celkový čas návštěvy (rozdíl mezi časem posledního a prvního zobrazení stránky v rámci návštěvy). Čas návštěvy tedy není přesný, neboť neuvažuje čas strávený na poslední zobrazené stránce. Singleton je rozpoznán jako zobrazení stránky ohraničené na časové ose více než třicetiminutovou prodlevou z obou stran (jinými slovy návštěva, obsahující jediné zobrazení stránky). Singletons mají v systému nulový čas a při následných výpočtech průměrného trvání návštěvy nejsou uvažovány.

#### 4.5.3 Prezentace výsledků

Jak bylo zmíněno v sekci 4.4.4, prezentační část MJmonitoru má 3 části — off-page, on-page a API. Vše je implementováno v rámci jedné aplikace, postavené na PHP frameworku Nette (viz sekce 4.3.1) s využitím architektury MVC. O práci s daty se starají základní dva modely, reprezentované statickými třídami *Analyzer* a *Statistics*. *Analyzer* provádí analýzy, popisované v předchozí kapitole, třída *Statistics* má na starosti finální úpravu dat pro prezentační účely.

Off-page část je řízena třídami *HomepagePresenter*, *VisitorsPresenter*, *SourcesPresenter* a *SettingsPresenter*. Prezenter se starají o přijetí požadavku od uživatele, získání potřebných dat z modelů a předání těchto dat šablonám. Šablony definují vzhled výsledné webové stránky s prezentací výsledků. Součástí aplikace je jednoduchý autentizační systém — aby si uživatel mohl prohlédnout výsledky, musí se přihlásit svým uživatelským jménem a heslem.

On-page část je na serveru řízena třídou *OnpagePresenter*, která spolupracuje s klientskou částí, tvořenou souborem *presenter.js* v jazyce JavaScript. Pokud chce uživatel využívat on-page prezentaci dat, musí do sledovaných webových stránek vložit prezentační kód (způsob vložení je podobný jako u měřicího kódu), jehož příklad si lze prohlédnout v ukázce 4.4.

```
<script type="text/javascript"
  src="http://umisteni.knihovny/presenter.js"></script>
<script type="text/javascript">
  try {
    mjmPresenter.registerShortcut();
  } catch (err) {}
</script>
```

Kód 4.4: Příklad prezentačního kódu ve sledované stránce



Prezentační kód pro svoje fungování vyžaduje, aby byl ve stránce vložen i měřicí kód, neboť využívá některé funkce ze souboru *tracker.js*. Běžný návštěvník webu o prezentačním kódu (stejně jako o měřicím kódu) neví, neboť na stránce není žádné viditelné tlačítko nebo odkaz pro spuštění on-page prezentace výsledků měření. Spuštění prezentace proběhne po stisku klávesové zkratky *Ctrl+M*. Pokud uživatel ve chvíli spuštění prezentace není přihlášen, je automaticky přesměrován na přihlašovací stránku off-page části, odkud je po úspěšném přihlášení přesměrován zpět na sledovanou stránku.

Výměna dat mezi klientskou a serverovou částí funguje způsobem, nastíněným v blokovém schématu na obrázku 4.1. Při spuštění on-page prezentace na webové stránce odešle skript *presenter.js* HTTP požadavek na serverovou část. Součástí požadavku je URL sledované stránky a časový interval, za který výsledky požadujeme. Serverová část tyto vstupní parametry zpracuje a na jejich základě vygeneruje kompletní HTML kód on-page panelu s požadovanými informacemi. Tento panel následně pošle zpět klientovi. Výměna dat mezi serverem a klientem probíhá metodou JSON-P, popisovanou v sekci 2.3.3.

## Kapitola 5

# Nástroj pro měření účinnosti internetové reklamy

Jak jednoduše měřit, zda se vyplatí investice do určitého druhu reklamy? Odpověď na tuto otázku není jednoduchá. V případě klasických reklamních médií (tisk, televize, outdoor) musíme účinnost reklamní kampaně často spíše odhadovat. Avšak v případě, že reklamní kampaň probíhá na internetu a inzerujeme v ní webové stránky, máme k dispozici pro určení její efektivity zajímavé možnosti.

Můžeme velmi přesně měřit, kolik potenciálních zákazníků naše reklama přilákala a v určitých případech můžeme mít k dispozici i údaje vypovídající o obratu (nebo přímo zisku), jenž byl dosažen zákazníky, získanými na základě konkrétní reklamní kampaně. Nejlépe se účinnost zjišťuje u výkonnostních reklamních kampaní, jejichž cílem je přímá objednávka zboží (typicky u internetových obchodů). Zde se dá velmi dobře vypočítat návratnost investice (tzv. ROI)<sup>1</sup>. O něco horší je situace u kampaní, zaměřených na lead generation<sup>2</sup> a ještě hůře se návratnost počítá u brandingových<sup>3</sup> kampaní. Tato práce však nemá za cíl rozebírat marketingovou stránku věci, pojďme se proto podívat na problém vyhodnocování účinnosti internetové reklamy z technického hlediska.

### 5.1 Metodika měření

Abychom mohli měřit účinnost internetové reklamní kampaně, potřebujeme znát některé její vlastnosti. V první řadě potřebujeme umět odlišit návštěvníky, pocházející z určité reklamní kampaně, od ostatních návštěvníků. Dále potřebujeme sledovat, zda se tito návštěvníci dostali až ke konverznímu cíli, o který nám jde. Potřebujeme znát zisk z konverzí, které provedli návštěvníci přichozí z dané kampaně. K tomu, abychom byli schopni určit návratnost investice do kampaně, potřebujeme znát kromě zisku také náklady, které byly na danou reklamní kampaň vynaloženy.

Jakmile dokážeme rozpoznat návštěvníky, kteří na web přišli díky sledované reklamní kampani, známe zisk, který tito návštěvníci na webu vygenerovali a zároveň známe náklady sledované kampaně, můžeme spočítat návratnost investice. V praxi bývá bohužel problém

---

<sup>1</sup>ROI (Return Of Investment), neboli návratnost investice je základním ukazatelem účinnosti internetové reklamy. Udává se v procentech a počítá se jako  $\text{zisk} / \text{investice} * 100$  [6]

<sup>2</sup>„Lead Generation je internetový způsob získávání leadů neboli kontaktů na potenciální klienty, kteří projeví skutečný a prokazatelný zájem pořídit si určitý produkt nebo službu.“ [20]

<sup>3</sup>Branding (od slova brand = anglicky značka) je komplex úkonů, vedoucích k vybudování úspěšné značky [14].

získat skutečné a přesné údaje jak o nákladech kampaně, tak o souvisejícím zisku, nicméně s určitými kompromisy se dá tento problém řešit.

Principem vyhodnocování účinnosti reklamních kampaní je tedy propojení dat o jejich nákladech s daty o zdrojích návštěvnosti a provedených konverzích.

### **Druhy internetových reklamních kampaní z hlediska systému zpoplatnění**

**PPC (Pay-Per-Click) kampaně** Zde je situace zdánlivě nejjednodušší. U těchto typů kampaní platíme až tehdy, pokud návštěvník klikne na reklamní odkaz a tím pádem se dostane na náš web. Pokud známe pevnou cenu za proklik z reklamní kampaně, znamená to, že známe i náklady na příchozího návštěvníka. V praxi však reálná cena za proklik nebývá pevně stanovená, a tak je potřeba ji aproximovat (např. použít odhad, nebo historický průměr reálných cen za proklik).

**CPM (Cost-Per-Mille) kampaně** U CPM kampaní se platí za určitý počet zobrazení reklamy (používá se též pojem *imprese*), nejčastěji je cena reklamy stanovena za tisíc impresí. Problémem zde může být zjištění skutečného počtu impresí, který potřebujeme pro výpočet nákladů. Řešení spočívá buď v aproximaci tohoto počtu (např. na základě počtu návštěv z reklamy a průměrného CTR<sup>4</sup>), nebo v generování reklamního odkazu nástrojem, který každé vygenerování započte jako jedno zobrazení (u textové reklamy toto řešení nebývá realizovatelné, je vhodnější spíše pro plošnou bannerovou reklamu).

**kampaně s platbou za časovou jednotku** U tohoto typu kampaní se platí pevná cena za určitý časový úsek, po který se reklama zobrazuje, často se platí za týdenní, měsíční, nebo roční období. Náklady jsou zde tedy časově závislé.

**affiliate kampaně** U affiliate kampaní se s poskytovatelem reklamního prostoru dělíme o zisk, vygenerovaný zákazníky, pocházejícími z dané kampaně (nejčastěji určitým procentem z obrátu).

Z uvedeného výčtu je patrné, že způsob výpočtu nákladů konkrétní reklamní kampaně závisí na způsobu, jakým je reklama zpoplatněna. Představme si, že máme k dispozici data o počtu návštěvníků, příchozích z dané kampaně, dále známe počet návštěvníků, kteří provedli konverzi a v neposlední řadě zisk, jenž tito návštěvníci vygenerovali. Všechna zmíněná data jsou omezena zadaným časovým intervalem. Jak vypočítáme náklady reklamní kampaně?

- U systému PPC vynásobíme počet návštěvníků, pocházejících z dané kampaně, cenou za proklik.
- U systému CPM si z dostupnými údaji nevystačíme, potřebujeme počet impresí. Ten můžeme zjišťovat buďto zmíněným generováním reklamního odkazu, nebo jej můžeme aproximovat např. podle počtu prokliků nebo doby trvání kampaně. Počet impresí je nakonec potřeba vynásobit cenou za impresi.
- U kampaní s platbou za časovou jednotku stanovíme náklady tak, že vynásobíme cenu za časovou jednotku poměrem délky zvoleného intervalu k této časové jednotce. Např. platíme-li 12 000 Kč za rok a sledujeme měsíční interval, náklady kampaně budou přibližně 1 000 Kč.

---

<sup>4</sup>CTR = Click Through Rate — tzv. míra prokliku, udává se v procentech a počítá se jako *počet prokliků / počet zobrazení \* 100* [6]

- U affiliate kampaní stanovíme náklady kampaně procentem z dosaženého zisku.

## 5.2 Implementace

Nástroj je implementován jako součást knihovny, představené v kapitole 4. Základem je na jedné straně měření konverzí a na straně druhé nastavení sledovaných reklamních kampaní. Vyhodnocení účinnosti reklamní kampaně získáme propojením údajů z těchto dvou částí.

### 5.2.1 Měření konverzí

V projektu rozlišujeme z hlediska sběru dat dva druhy konverzních cílů:

**dynamické** Pro jejich zaznamenání je potřeba vložit na stránku, která následuje po dosažení konverzního cíle, speciální prvek měřicího kódu — zavolat metodu *mjm.trackConversion()*. Této metodě předáme tři parametry: zkratku konverzního cíle, označení konkrétní konverze a hodnotu konverze. Typickým příkladem je objednávka v internetovém obchodu, u které jako označení konkrétní konverze uvedeme číslo objednávky a jako hodnotu konverze uvedeme zisk z této objednávky.

**statické** Statické konverzní cíle mají pevně nastavenou hodnotu a jejich rozpoznávání funguje na základě nastavené cílové URL adresy. Příkladem takového cíle je odeslání kontaktního formuláře na webu.

Konverze dynamických cílů zaznamenává serverový skript *tracker.php* do zvláštní kolekce. Konverze statických cílů se nikam speciálně nezaznamenávají, jejich analýza probíhá zpětně nad kolekcí *pageViews*.

### 5.2.2 Rozpoznávání návštěvníků z reklamní kampaně

Základem rozpoznávání, kteří návštěvníci přišli na web z určité reklamní kampaně, tkví v nastavení „poznávacích znaků“ URL zdroje. V systému lze tvar zdrojové URL adresy definovat buď pomocí regulárního výrazu, nebo jednodušeji specifikací prvků, jež má zdrojová URL adresa obsahovat. Dále je již princip shodný s rozpoznáváním ostatních zdrojů návštěvnosti. Rozsah a charakter této písemné zprávy neumožňuje zde podrobně popsat konkrétní použité algoritmy, případné zájemce proto mohou odkázat na zdrojové kódy MJmonitoru, které jsou k nalezení na přiloženém CD (viz [A](#)).

## Kapitola 6

# Testování v reálném prostředí

Navržené řešení se dá testovat různými způsoby. Bylo by jistě možné např. vytvořit testovací skript, který by navštěvoval měřený web z různých prohlížečů, pomocí mazání cookies by simuloval více návštěvníků, apod. Nicméně jako nejlépe vypovídající se mi jeví praktické testování na reálném webu, kde by bylo možné výsledky porovnat s některým z existujících nástrojů.

Pro účely testování projektu jsem vybral dva weby zcela odlišného charakteru a návštěvnosti. Na obou webech jsou k dispozici i jiné analytické nástroje, takže existuje možnost porovnání. Testování v reálném prostředí jsem prováděl již od samotného počátku vývoje, což se ukázalo jako dobrý tah, neboť mi to umožnilo odhalit velké množství skrytých a záludných chyb.

Pro účely této zprávy jsem jako testovací období vybral interval od 4. 5. 2011 do 11. 5. 2011.

### 6.1 Web s vyšší návštěvností

Jako web s vyšší návštěvností jsem vybral internetový obchod [www.hunting-shop.cz](http://www.hunting-shop.cz). Návštěvnost tohoto e-shopu se stabilně pohybuje v rozmezí 500–1000 návštěv za den. Je zde možnost praktického měření konverzí, jejichž cílem je provedení objednávky. Web využívá nástroje Google Analytics a AWStats.

Srovnání výsledků, naměřených MJmonitorem a těmito nástroji, je v grafech na obrázku 6.1 a v tabulce 6.1. Z grafů je patrné, že v počtu návštěv a v počtu zobrazení stránek se nástroje sice liší v absolutních počtech, ale výkyvy denních výsledků odpovídají. Nejvyšší počet návštěv naměřil nástroj AWStats, nejnižší Google Analytics, čísla MJmonitoru se pohybují mezi výsledky z ostatních dvou nástrojů. Odchylka se na obě strany pohybuje kolem pěti procent. U počtu zobrazení stránek je odchylka již výraznější, pohybuje se přibližně mezi deseti a třiceti procenty. Nejvyšší počet zobrazení stránek naměřil opět nástroj AWStats, nejnižší čísla vykázal MJmonitor.

Není překvapivé, že AWStats naměří vyšší počty, neboť tento nástroj využívá principu analýzy logů webového serveru. V serverovém logu jsou zaznamenána skutečně všechna zobrazení stránek, zatímco u nástrojů na principu měřicího kódu se některá zobrazení z různých důvodů nemusí zaznamenat (např. uživatelův prohlížeč nepodporuje JavaScript, nebo bylo zobrazení stránky příliš krátké a uživatel přešel na jinou stránku dříve, než se stačila odeslat data na server). Důvodů, proč MJmonitor naměří ještě nižší počet zobrazení, než Google Analytics, který pracuje na podobném principu, může být několik. Google Analytics může

Název zdroje / doména	Počet přístupů	
	MJM	GA
přímá návštěvnost	2163	844
Seznam.cz	1655	1857
Google.cz	821	165
Zbozi.cz	396	353
Google.sk	289	36
bazar.hunting-shop.cz	141	189
hobby-army.hyperinzerce.cz	133	97
Google.com	93	17

Tabulka 6.1: Srovnání nejčastějších zdrojů návštěvnosti webu hunting-shop.cz podle nástrojů MJmonitor a Google Analytics

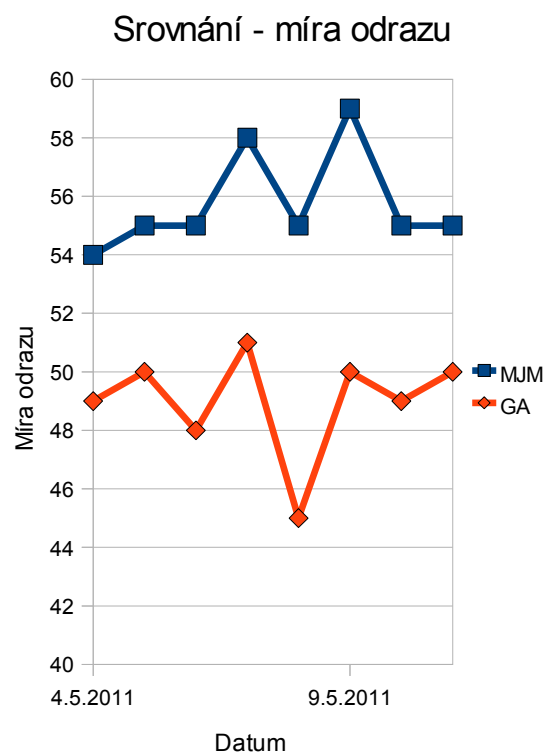
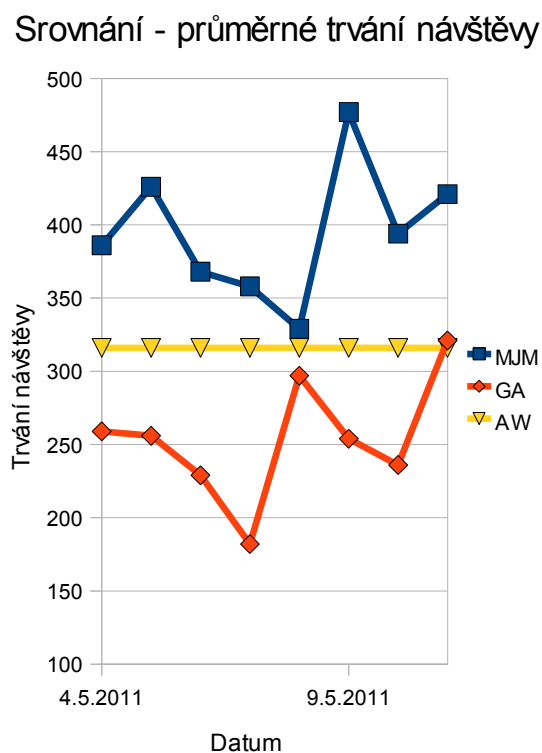
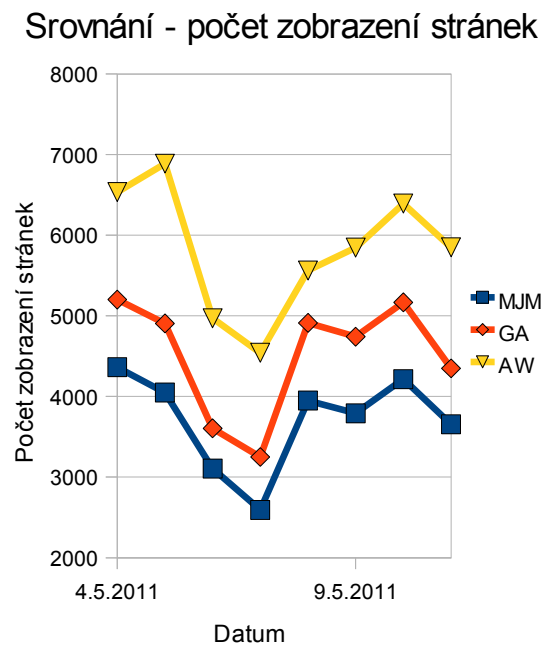
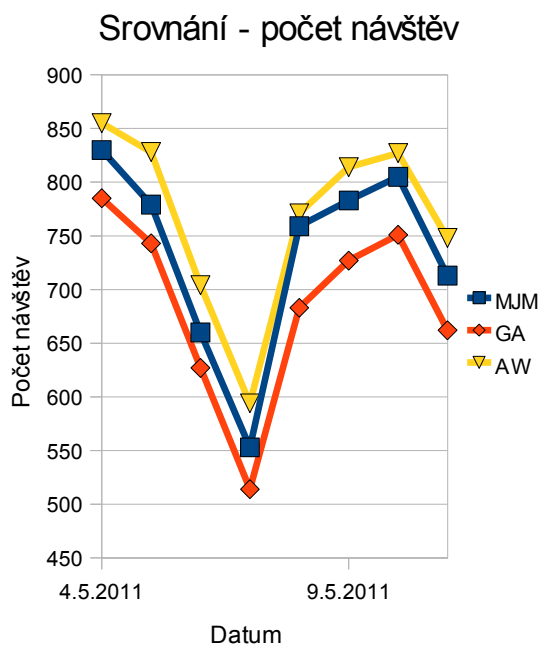
mít např. kvalitnější měřicí kód, který se dokáže vykonat ve větším procentu případů, než sběrný kód MJmonitoru. Roli hraje rovněž to, že MJmonitor ignoruje uživatele s vypnutými cookies, aby nedocházelo ke zkreslení výsledků. Metodika Google Analytics v tomto případě není známa, nicméně lze se domnívat, že je použita jiná (byť méně spolehlivá) forma identifikace návštěvníka a do výsledků se tak počítají i uživatelé bez cookies.

Překvapivé jsou výsledky měření zdrojů návštěvnosti (viz tabulka 6.1). Ty na první pohled vycházejí u MJmonitoru a Google Analytics zcela rozdílně (AWStats neumožňuje vypsat výsledky za ručně zvolený časový úsek, proto v tabulce chybí). Pokud výsledky srovnáme podrobněji, zjistíme, že výsledky u některých zdrojů srovnávat příliš nelze. Největší rozdíly jsou u zdrojů Google.cz, Google.sk a Google.com, kde Google Analytics vypisuje mnohem menší čísla, než MJmonitor. V Google Analytics je však navíc obecnější zdroj *google / organic*, vykazující 1298 návštěv. Pokud bychom tento počet rozdělili mezi tři nejvýznamnější Google-zdroje v MJmonitoru, výsledný rozdíl by již byl velmi malý. Rozdíly u ostatních zdrojů návštěvnosti mohou být dány rozdílnou metodikou. MJmonitor udává počet přístupů z daného zdroje, zatímco Google Analytics udává počet návštěv. Pokud tedy uživatel navštíví během jedné návštěvy web z několika zdrojů, Google Analytics pravděpodobně započítá pouze ten první. Dalším důvodem rozdílů může být odlišné provedení rozpoznávání jednotlivých zdrojů.

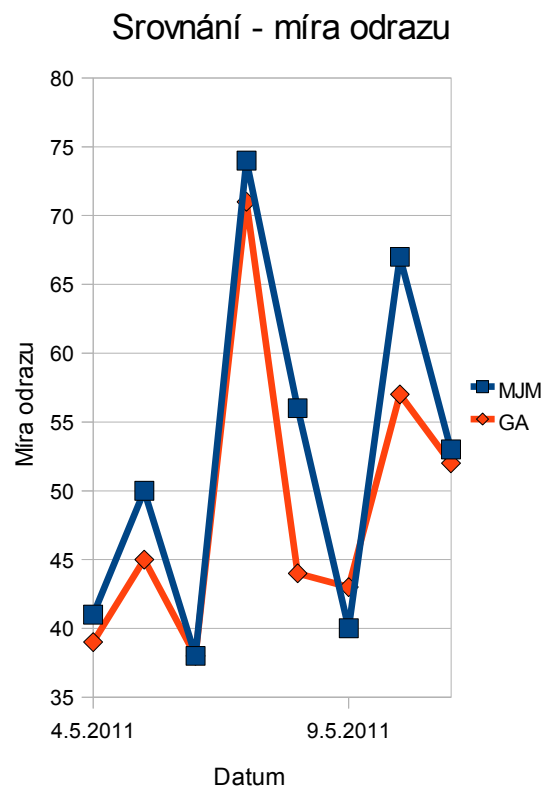
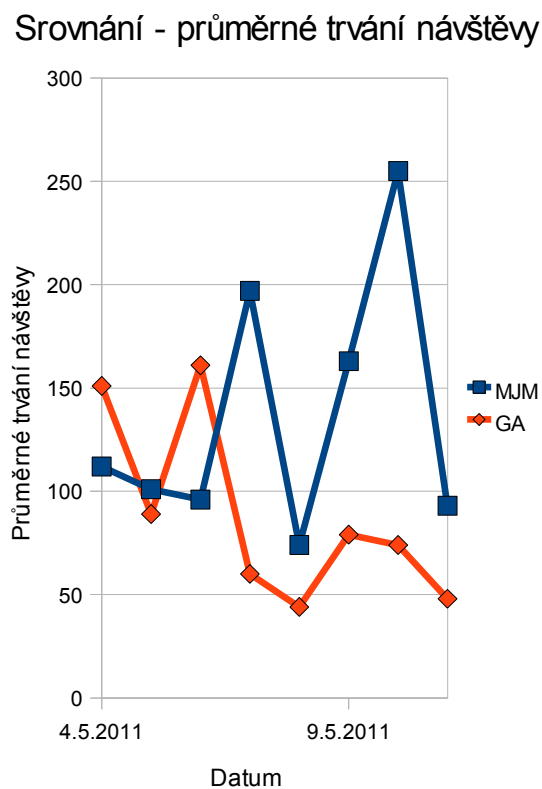
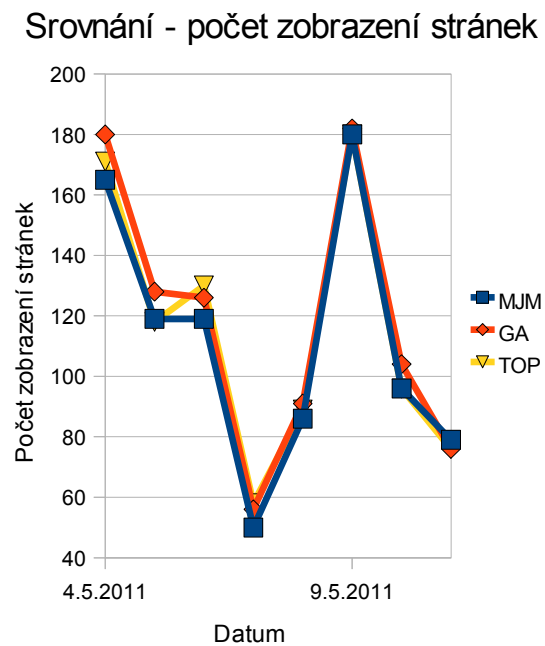
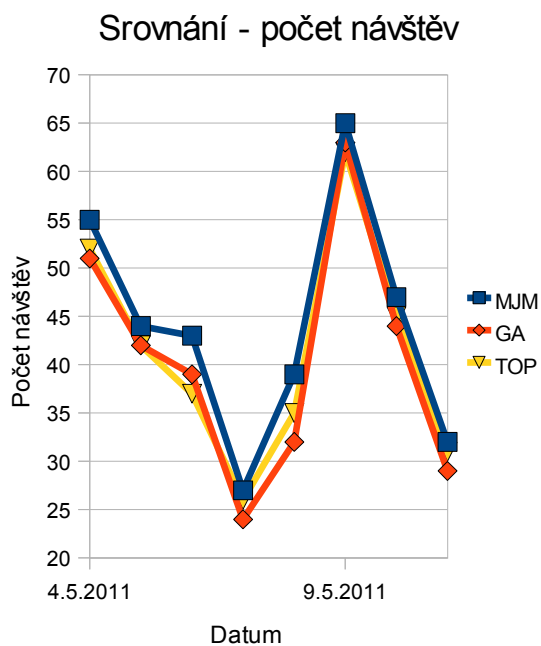
## 6.2 Web s nižší návštěvností

Jako web s nižší návštěvností byla vybrána prezentace [www.mrzena.eu](http://www.mrzena.eu). Jedná se o jednoduchý prezentační web živnostníka, podnikajícího v oboru kovovýroby a dřevěných staveb. Návštěvnost se zde pohybuje v rozmezí 20–80 návštěv denně. Konverzním cílem je v rámci tohoto webu odeslání kontaktního formuláře. Web využívá nástroje Google Analytics a TopList.

Srovnání výsledků měření MJmonitoru a ostatních nástrojů si lze prohlédnout v grafech na obrázku 6.2 a v tabulce 6.2. Rozdíly jsou zde podobného charakteru, jako u webu hunting-shop.cz v předchozí kapitole, kde jsou i vysvětleny.



Obrázek 6.1: Soubor grafů: srovnání naměřených hodnot na webu hunting-shop.cz prostřednictvím nástrojů MJmonitor, Google Analytics a AWStats



Obrázek 6.2: Soubor grafů: srovnání naměřených hodnot na webu mrzena.eu prostřednictvím nástrojů MJmonitor, Google Analytics a TopList



Název zdroje / doména	Počet přístupů	
	MJM	GA
Seznam.cz	96	206
přímá návštěvnost	43	41
Google.cz	11	7
www.firmy.cz	7	10

*Tabulka 6.2:* Srovnání nejčastějších zdrojů návštěvnosti webu mrzena.eu podle nástrojů MJmonitor a Google Analytics

## Kapitola 7

### Závěr

V rámci této bakalářské práce byla navržena a implementována knihovna, umožňující monitorování návštěvníků webových stránek, včetně sledování jejich zdrojů a chování. Knihovna je realizována takovým způsobem, že ji lze použít bez jakékoliv další nadstavby i jako plnohodnotný samostatný nástroj pro webovou analytiku. Specifickou částí knihovny je nástroj pro vyhodnocení účinnosti internetové reklamy.

Implementovaný systém byl otestován v prostředí reálných webových stránek. Získané výsledky byly vyhodnoceny a porovnány s výstupy jiných nástrojů pro webovou analytiku.

Za přínos projektu považuji zvolený způsob sběru a ukládání dat, který umožňuje snadnou rozšiřitelnost a s drobnými úpravami i škálovatelnost pro případ vysoké zátěže. Za přínosnou považuji též zvolenou formu prezentace dat, zejména v rámci on-page části, díky které uživatel získává přehled o pohybu návštěvníků přímo v reálném prostředí a snáze si tak uvědomí podstatné souvislosti.

Výhodou může být možnost instalace knihovny na privátní server. V současné době nejpoužívanější nástroje pro webovou analytiku fungují zpravidla formou hostované služby, tzn. data jsou umístěna na serveru poskytovatele. To může majiteli sledovaného webu vadit, pokud má strach ze zneužití naměřených dat třetí stranou.

Vzniklý nástroj plánuji nadále rozvíjet. Díky zvolené systémové i implementační architektuře lze snadno rozšiřovat možnosti nástroje — např. analyzovat shromážděná data z jiných úhlů pohledu, nebo přidávat další typy sledovaných dat. Jako rozšíření se nabízí například sledování pozic kliknutí uživatele a následná tvorba heatmap jednotlivých webových stránek (podobně, jako to umí nástroje mYx a CrazyEgg, popsané v kapitole 3). Bylo by též možné do určité míry sledovat u uživatelů pohyby myši a spolu s heatmapami tak využít nástroj k testování použitelnosti webových stránek.

# Literatura

- [1] Hummingbird: Real Time Web Traffic Visualization [online].  
<http://projects.nuttnet.net/hummingbird/>.
- [2] MapReduce [online]. <http://www.mongodb.org/display/DOCS/MapReduce>.
- [3] NetMonitor [online]. <http://www.netmonitor.cz/>.
- [4] Bryzek, M.: Real time ecommerce analytics with MongoDB at Gilt Groupe [online].  
<http://www.slideshare.net/mongosf/real-time-ecommerce-analytics-with-mongodb-at-gilt>  
2010-04-30 [cit. 2011-05-11].
- [5] Eckersley, P.: How Unique Is Your Web Browser?  
<http://panopticlick.eff.org/browser-uniqueness.pdf>, 2009.
- [6] Janouch, V.: *Internetový marketing: Prosaďte se na webu a sociálních sítích*.  
Computer Press, 2010, ISBN 978-80-251-2795-7.
- [7] Jeffrey Dean and Sanjay Ghemawat: MapReduce: Simplified Data Processing on Large Clusters.  
[http://static.googleusercontent.com/external\\_content/untrusted\\_dlcp/labs.google.com/](http://static.googleusercontent.com/external_content/untrusted_dlcp/labs.google.com/)  
2004.
- [8] Kolektiv autorů: The Official WAA Definition of Web Analytics [online].  
<http://www.webanalyticsassociation.org/?page=aboutus>.
- [9] Kolektiv autorů: NAVRCHOLU.cz: Lidé se JavaScriptu a cookies nebojí [online].  
2007-04-24 [cit. 2011-04-13].
- [10] Kolektiv autorů: Web analytics [online].  
[http://en.wikipedia.org/wiki/Web\\_analytics](http://en.wikipedia.org/wiki/Web_analytics), 2011-03-27 [cit. 2011-04-12].
- [11] Kolektiv autorů: Analytics and Tracking Distribution [online].  
<http://trends.builtwith.com/analytics>, 2011-04-18 [cit. 2011-04-18].
- [12] Kolektiv autorů: Usage of traffic analysis tools for websites [online].  
[http://w3techs.com/technologies/overview/traffic\\_analysis/all](http://w3techs.com/technologies/overview/traffic_analysis/all), 2011-04-18  
[cit. 2011-04-18].
- [13] Kolektiv autorů: MongoDB Native Driver [online].  
<http://cz.php.net/manual/en/book.mongo.php>, 2011-05-06 [cit. 2011-05-11].
- [14] Kolektiv autorů: Obchodní značka [online].  
[http://cs.wikipedia.org/wiki/Brand\\_%28obchod%29#Branding](http://cs.wikipedia.org/wiki/Brand_%28obchod%29#Branding), 2011-05-11  
[cit. 2011-05-14].

- [15] Kumar, J.: Document oriented data stores [online]. 2009-08-11 [cit. 2011-05-11].
- [16] Mooberry, J.: MongoDB and Symfony? Yes! Part 1: Inserts [online]. 2009-08-22 [cit. 2011-05-11].
- [17] Němec, R.: Výhody a nevýhody Google Analytics [online].  
<http://webova-analytika.robertnemec.com/google-analytics-vyhody-nevyhody/>,  
2009-09-01 [cit. 2011-04-18].
- [18] Petrášek, M.: Přednosti poradenských služeb založených na Yahoo Web Analytics [online].  
<http://www.slideshare.net/adobryweb/yahoo-web-analytics-martin-petrek>,  
2009.
- [19] Videla, A.: Benchmarking MongoDB VS. Mysql [online]. 2009-07-02 [cit. 2011-05-11].
- [20] Čurda, P.: Lead generation marketing - Srovnej nabídky na trhu, ale ať jsme nejlevnější [online]. <http://www.lupa.cz/clanky/lead-generation-marketing/>,  
2010-08-17 [cit. 2011-05-14], iSSN 1213-0702.

## Dodatek A

### Obsah CD

**adresář** *mjmonitor* Zdrojové kódy knihovny/nástroje MJmonitor. Pro instalaci na server stačí zkopírovat obsah tohoto adresáře do kořenového adresáře cílové domény nebo subdomény, na níž má knihovna běžet a nastavit oprávnění, umožňující PHP skriptům zápis do adresářů *temp* a *log*.

**adresář** *mjmonitor-api* Programová dokumentace knihovny/nástroje MJmonitor.

**adresář** *tex* Zdrojové kódy této písemné práce.

**soubor** *xjelic00.pdf* Elektronická podoba této písemná práce ve formátu PDF.

## Dodatek B

# Požadavky pro chod knihovny/nástroje MJmonitor

Pro správnou funkčnost MJmonitoru je zapotřebí databáze MongoDB ve verzi 1.8 nebo vyšší a interpret PHP ve verzi 5.3 nebo vyšší s nainstalovaným rozšířením Mongo.